

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-149450

(43)Date of publication of application : 24.05.2002

---

(51)Int.Cl. G06F 12/00  
G06F 17/30

---

(21)Application number : 2000-348963 (71)Applicant : RICOH CO LTD

(22)Date of filing : 16.11.2000 (72)Inventor : ASADA KAZUSHIGE  
TAKEGAWA HIROSHI  
HIRAOKA TAKUYA  
IKEDA TETSUYA  
MAEDA KAORU

---

## (54) METHOD AND DEVICE FOR MANAGING DATABASE

### (57)Abstract:

PROBLEM TO BE SOLVED: To make schemer information changeable even during the data processing of a database is being executed.

SOLUTION: This database management device is provided with a data base file 131 constituted of a plurality of data information filea schemer file 133 constituted of a plurality of schemer information filesa lock file 132 capable of storing a lock mode corresponding to every data processing request classification for every data information file and schemer information filean inquiry optimizing means 125 for checking the competing state of the generated data processing requests by referring to the lock mode of each data information file and schemer information file stored in the lock file 132 every time a new data professing request is generated and for generating the execution procedure of the optimal data processingand an inquiry executing means 126 for setting the lock mode in the lock file 132 prior to the execution of the requested data processing according to the execution procedure and for releasing the set lock mode when the data processing is ended.

---

## CLAIMS

---

[Claim(s)]

[Claim 1]Have the following and two or more lock modes for preventing competition of

a data-processing demand to said data information file and said schema information file are provided according to classification of said data-processing demandBy making this lock mode set up corresponding to classification of said data-processing demand for said every data information file and said every schema information fileOnly not only in parallel operation of two or more data processing which receives said data information fileA database management method making possible parallel operation of two or more data processing which receives parallel operation of two or more data processing which receives said data information file and said schema information fileand said schema information file.

A database file which consists of two or more data information files.

A schema file which consists of two or more schema information files which define each data structure of this data information file.

[Claim 2]In a database management method according to claim 1said data information file and said schema information fileIt is constituted by file format of a tabular formatrespectivelyand setting out of said lock modeBy making it set it as each front unit which constitutes said data information file and said schema information file corresponding to classification of said data-processing demandOnly not only in parallel operation of two or more data processing which receives said data information file which constitutes a tableA database management method making possible parallel operation of two or more data processing which receives said schema information file which constitutes two or more parallel operation and tables of data processing which receive said data information file which constitutes a tableand said schema information file.

[Claim 3]In a database management method according to claim 2for every row unit which is every line unit and a data attribute which are the data record used as a component of said data information file of a tabular formatand each file of said schema information file. By making said lock mode set up corresponding to classification of said data-processing demandOnly not only in parallel operation of two or more data processing which receives said data information fileA database management method making possible parallel operation of two or more data processing which receives parallel operation of two or more data processing which receives said data information file and said schema information fileand said schema information file.

[Claim 4]In a database management method according to any one of claims 1 to 3In a case where said data information file has index data which performs correlation between said two or more data information filesand said schema information file has the index information which defines an index relation of this index dataBy making said lock mode set up corresponding to classification of said data-processing demand for said every index information in said every index data in said data information fileand said schema information fileA database management method making possible parallel

operation of 1 thru/or two or more data processing to said index information in said schema information file and 1 thru/or two or more data processing to said corresponding index data in said data information file which this index information defines.

[Claim 5] In a database management method according to any one of claims 1 to 4 When it has a trigger and an assertion function to perform automatically data processing beforehand defined when conditions defined beforehand were satisfied a data-processing demand by this trigger and an assertion function is satisfied By making said lock mode set it as said data information file and said schema information file 1 thru/or two or more data processing to said trigger and said data information file based on an assertion function or said schema information file A database management method making possible parallel operation with 1 thru/or two or more data processing to other said data information files based on a data-processing demand or said schema information files other than said trigger and an assertion function.

[Claim 6] In a database management method according to any one of claims 1 to 5 Based on a setting condition of a lock mode set as said data information file or said schema information file A database management method characterized by making the order of the real way of this data processing optimize so that execution time of data processing to said data information file or said schema information file may serve as the shortest.

[Claim 7] In a database management device which has a database file which consists of two or more data information files and a schema file which consists of two or more schema information files which define each data structure of this data information file Two or more lock modes for preventing competition of a data-processing demand to said data information file and said schema information file A lock file in which said lock mode which is provided according to classification of said data-processing demand and is set up corresponding to classification of said data-processing demand is made to store for said every data information file and said every schema information file Each time when a data-processing demand to said data information file or said schema information file occurs Said lock mode stored in schema information stored in said schema information file and said lock file is referred to Check a race condition of said generated data-processing demand and while performing data processing required as an inquiry optimizing means which generates the order of the real way of optimal data processing according to this generated order of the real way execution of this data processing is preceded A database management device having an inquiry execution means of which said lock mode made to store in said lock file is made to cancel when said lock mode corresponding to said generated data-processing demand is made to store in said lock file and execution of said data processing is completed.

[Claim 8] It has a database file which consists of two or more data information files and a schema file which consists of two or more schema information files which define

each data structure of this data information fileTwo or more lock modes for preventing competition of a data-processing demand to said data information file and said schema information file are provided according to classification of said data-processing demandBy making this lock mode set up corresponding to classification of said data-processing demand for said every data information file and said every schema information fileOnly not only in parallel operation of two or more data processing which receives said data information fileA database management method which makes possible parallel operation of two or more data processing which receives parallel operation of two or more data processing which receives said data information file and said schema information fileand said schema information fileA recording medium which is made to record as a program for making a computer carry out and in which computer reading is possible.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention]this invention -- a database management system (DBMS) -- in a Relational Database Management System (RDBMS) especiallyIt is related with the database management method which enables reconstruction of a database including change (namelychange of schema information) of the data structure of the data information file which constitutes a databasemaking employment of database service continue without installing a spare database. It is related with the program recording medium [ the database management device which realizes this database management methodand the computer on which this database management method was made to record as a program performed by computer ] which can be read.

[0002]

[Description of the Prior Art]Conventionallyin the database management system applied to online processing systems some art for making a database reconstruct has been indicatedwithout interrupting an on-line service. Herereconstruction of a database changes the structure of the old database hereand it is made to change it into the structure of a new database. For examplewhile the digit number of a zip code changes the size of the data area of triple figures which has memorized the zip code in an address-book-data information file in connection with having been changed into 7 figures from triple figures and makes it extend to the new data area of 7 figuresThe address-book-data information file which made a seven-digit new zip code store in the data area made to extend is made to create.

[0003]For examplethere is database restructuring art currently indicated in JPH5-233408A "migration method in online processing systems" to realize reconstruction of this databasewithout interrupting an on-line service. The on-line service was freely

made to continue by a piece system in this gazette using the data file memory storage of a duplicate configuration. Transition processing to the data information file of new composition is performed on another another system. At the time of transition processing completion it is going to stop the downtime of an on-line service to the minimum by stopping an on-line service once changing an on-line system to another system which shifted to the data information file of new composition and making an on-line service resume by a new system.

[0004] Using the data file memory storage of this duplicate configuration at the time of a change in an on-line system, a doubleless synchronized operation is performed, a change is realized in an instant without stopping an on-line service, and the on-line service database restructuring art in which it is not interrupted is also proposed.

[0005] However, since a duplicate configuration will be the requisite also in a stationary state, such database system needs two memory storage which completely stores the same data information file and program file. The various special control facilities for controlling a duplicate configuration will also be needed, and it will be an expensive system configuration. That is, it cannot be made to realize in the database system which consists of the usual single data file memory storage.

[0006] On the other hand, there is no discontinuation of an on-line service also in the database system of a single system which does not have a reserve system, and there are some which are indicated by JPH7-244603A "database restructuring method" as art of realizing reconstruction of a database.

[0007] An abstract data-access means by which the method currently indicated in this gazette accesses the accessing means to the data information file on data file memory storage without being conscious of the place and size on physical data file memory storage [after it makes it separate into two with real data-access means to access consciously and a real data-access means reconstructs a data structure] have a conversion method of the data information file between old and new as an abstract data-access means -- let it be a plug to update a data information file even to which an old and new data information file with the operating status of an on-line system possible by closing.

[0008] Also in JPH7-325744A "database form conversion method and device" on-line service uninterrupted database restructuring in the single data file memory storage which does not have a reserve system is made possible. When the demand of the conversion process of the data structure of a data information file is generated, the method currently indicated in this gazette Once freeze the renewal of the original data information file and it is made to record on the log file which was able to establish independently the renewal record which should be given to this data information file one by one. On the other hand, after making the data structure of the original data information file change to a desired data structure, it is copied one by one from the old data information file making it change into a new data format. When conversion of all the data structures is completed, about the data currently recorded on the log file it

changes into a new data format and re-creates and also the overwrite copy is carried out.

[0009] Also in JPH08-077050A "database restructuring device and database restructuring method" It has a log file in which the update history information of the old table for reconstruction and the old table is stored up. After replacing all the information on the old table by a new table and copying it based on the update history information accumulated in the log file, the uninterrupted reconfiguration method made to reflect in a new table about the information on the old table after updating is proposed.

[0010]

[Problem(s) to be Solved by the Invention] However, in a Prior art with the single database system which does not have a reserve system, receiving the data-processing demand to the data information file which has entity data of a database and making a data information file update. Until it cannot make data reconstruction perform at the time but the change processing (namely data reconstruction processing) of the structure of this data information file is completed although the structure of a data information file is simultaneously made to change. For example, it is necessary to make the information about data processing (data update process) of the data information file concerning the received data-processing demand save at the log file etc. which are made to prepare separately.

[0011] Therefore, in the on-line system with which a data-processing demand occurs in large quantities and needs to advance on-line processing promptly simultaneously using the updated latest data. Even if the situation where on-line operation must be carried out where employment of database processing is delayed to near the allowable limit in connection with making it once interrupted arises or it makes on-line operation continue. That a lot of log files are needed and when the worst the situation of it becoming impossible to acquire the latest data about the data information file needed for on-line operation in allowable delay time and stopping the on-line operation itself may arise.

[0012] In addition to the lock mode for the prevention from competition to two or more data-processing demands which this invention is made in view of this problem and are related with a data information file. By providing the lock mode for the prevention from competition also to two or more data-processing demands related with the schema information file which defines the data structure of this data information file. Let it be a plug to make parallel two or more data processing which receives a data information file and two or more data processing (namely data reconstruction processing) which makes the structure of a data information file change and to carry out simultaneous operation possible.

[0013]

[Means for Solving the Problem] A database file which an invention of Claim 1 becomes from two or more data information files. It has a schema file which consists of

two or more schema information files which define each data structure of this data information file Two or more lock modes for preventing competition of a data-processing demand to said data information file and said schema information file are provided according to classification of said data-processing demand By making this lock mode set up corresponding to classification of said data-processing demand for said every data information file and said every schema information file Only not only in parallel operation of two or more data processing which receives said data information file Parallel operation of two or more data processing which receives parallel operation and said schema information file of two or more data processing which receives said data information file and said schema information file is made possible.

[0014] In a database management method according to claim 1 an invention of Claim 2 Said data information file and said schema information file are constituted by file format of a tabular format respectively By making setting out of said lock mode set it as each front unit which constitutes said data information file and said schema information file corresponding to classification of said data-processing demand Only not only in parallel operation of two or more data processing which receives said data information file which constitutes a table Parallel operation of two or more data processing which receives said schema information file which constitutes two or more parallel operation and tables of data processing which receive said data information file which constitutes a table and said schema information file is made possible.

[0015] In a database management method according to claim 2 an invention of Claim 3 For every row unit which is every line unit and a data attribute which are the data record used as a component of said data information file of a tabular format and each file of said schema information file. By making said lock mode set up corresponding to classification of said data-processing demand Only not only in parallel operation of two or more data processing which receives said data information file Parallel operation of two or more data processing which receives parallel operation and said schema information file of two or more data processing which receives said data information file and said schema information file is made possible.

[0016] In a database management method according to any one of claims 1 to 3 an invention of Claim 4 In a case where said data information file has index data which performs correlation between said two or more data information files and said schema information file has the index information which defines an index relation of this index data By making said lock mode set up corresponding to classification of said data-processing demand for said every index information in said every index data in said data information file and said schema information file Parallel operation of 1 thru/or two or more data processing to said index information in said schema information file and 1 thru/or two or more data processing to said corresponding index data in said data information file which this index information defines is made possible.

[0017] In a database management method according to any one of claims 1 to 4 an invention of Claim 5 When it has a trigger and an assertion function to perform

automatically data processing beforehand defined when conditions defined beforehand were satisfied a data-processing demand by this trigger and an assertion function is satisfied By making said lock mode set it as said data information file and said schema information file 1 thru/or two or more data processing to said trigger and said data information file based on an assertion function or said schema information file Parallel operation with 1 thru/or two or more data processing to other said data information files based on a data-processing demand or said schema information files other than said trigger and an assertion function is made possible.

[0018] In a database management method according to any one of claims 1 to 5 an invention of Claim 6 Based on a setting condition of a lock mode set as said data information file or said schema information file The order of the real way of this data processing is made to optimize so that execution time of data processing to said data information file or said schema information file may serve as the shortest.

[0019] A database file which an invention of Claim 7 becomes from two or more data information files Two or more lock modes for preventing competition of a data-processing demand to said data information file which has a schema file which consists of two or more schema information files which define each data structure of this data information file and said schema information file A lock file in which said lock mode which is provided according to classification of said data-processing demand and is set up corresponding to classification of said data-processing demand is made to store for said every data information file and said every schema information file Each time when a data-processing demand to said data information file or said schema information file occurs Said lock mode stored in schema information stored in said schema information file and said lock file is referred to An inquiry optimizing means which checks a race condition of said generated data-processing demand and generates the order of the real way of optimal data processing While performing demanded data processing according to this generated above said lock mode corresponding to said generated data-processing demand is made to store in said lock file in advance of execution of this data processing When execution of said data processing is completed it has an inquiry execution means of which said lock mode made to store in said lock file is made to cancel.

[0020] A database file which an invention of Claim 8 becomes from two or more data information files It has a schema file which consists of two or more schema information files which define each data structure of this data information file Two or more lock modes for preventing competition of a data-processing demand to said data information file and said schema information file are provided according to classification of said data-processing demand By making this lock mode set up corresponding to classification of said data-processing demand for said every data information file and said every schema information file Only not only in parallel operation of two or more data processing which receives said data information file A database management method which makes possible parallel operation of two or more



data processing which receives parallel operation of two or more data processing which receives said data information file and said schema information file and said schema information file. It is made to record as a program for making a computer carry out.

[0021]

[Embodiment of the Invention] About the embodiment of this invention a figure is used and explained below. Drawing 2 is a block line block diagram showing the hardware constitutions of the database system concerning one working example of this invention. The communication network 40 is passed in drawing 2 and it is the terminal 1. Interconnection of 20—terminal N 30 and the server 10 which is building database DB130 is carried out. Here the user using database DB130 is the terminal 1. If the inquiry demand (namely data-processing demand) to 20—the data information file of the request from terminal N 30 is transmitted via the communication network 40. The server 10 searches database DB130, performs processing to the demanded data and returns it to the terminal which asked the data processing result and transmitted the demand.

[0022] It is the terminal 1 here. 20—terminal N 30 are constituted by computers such as PC. Namely each terminal is provided with CPU 21—31, the memories 22—32 and the hard disks 23—33 and respectively in each memories 22—32. The inquiry request input means 221—321 for emitting the inquiry demand to database DB130 on the server 10 respectively. The inquiry result display means 222—322 for receiving and displaying an inquiry result etc. can be stored and it can be made to perform as a program by each CPU 21—31. The keyboards 24—34 and pointing device for inputting the inquiry demand to database DB130 into each terminal respectively. (For example mouse) The displays 26—36 for 25—35 being equipped and displaying each inquiry result of database DB130 are equipped.

[0023] The communication network 40 is the terminal 1. Constitute 20—the transmission line for carrying out interconnection of terminal N 30 and the server 10 and generally the cable realizes and it is the terminal 1 by communications protocols such as TCP/IP. Information exchange of a between [ 20—terminal N 30 and the server 10 ] is made. However terminal 1. The communication network 40 may be built by 20—the wireless network using not only a cable cable but the electric wave and broadcast wave for wireless LAN if the communications protocol of terminal N 30 and the server 10 could be coincided.

[0024] The server 10 passes the communication network 40 and is the terminal 1. Inquiry demand SQL (Structured Query Language) of 20—database DB130 that were inputted from terminal N 30 i.e. a data-processing demand is received. Terminal 1 which performed data processing according to this inquiry demand and emitted this inquiry demand for the inquiry result. They are 20—a computer for returning terminal N 30 via the communication network 40. Namely the keyboard 14 for providing an input output means for the server 10 to manage a man machine interface, the pointing device 15 Or

while having display 16 grade it has CPU 11 the memory 12 and the hard disk 13. It consists of a stored-program-control system which can perform database management system DBMS 120 which manages various kinds of controlling functions to database DB 130 stored on the hard disk 13 as a program on the memory 12.

[0025] Next, the detailed block configuration of database management system DBMS 120 carried on the memory 12 of the server 10 and database DB 130 which are stored in the hard disk 13 is explained using drawing 1. It is a block line block diagram showing various kinds of processings (MEANS) which constitute database management system DBMS 120 which drawing 1 requires for this invention and various kinds of information files which constitute database DB 130 used as the processing object of this processing (MEANS) here. Database DB 130 consists of the database file 131, the lock file 132 and the schema file 133. The database file 131 is an information file group which consists of two or more data information files accumulated by the file format (for example, tabular format) which has a data structure separately defined considering the data object used as the substance of a database as the schema (schema).

[0026] The structure of each data in each data information file itself and this data information file form, relation, restriction, etc. are described in the schema (schema) here. The structure (it is hereafter written as the data structure of a data information file) of each data which is each component of the data information file itself or this data information file is defined. The information about this schema is a predetermined file format (for example, tabular format) as a schema information file is made to correspond to a data information file and is created. Two or more created schema information files are accumulated as the schema file 133 into the database 130. The structure (it is hereafter written as the data structure of a schema information file) of the schema information used as this schema information file itself or each component of a schema information file is also saved as a schema information file.

[0027] The lock file 132 is an information file which stores the lock-mode information for enabling exclusive control processing which controls the race condition of the data-processing demand to the data information file in the database file 131 and the schema information file in the schema file 133. It comprises two or more lock information files provided corresponding to the data information file in the database file 131 and each schema information file in the schema file 133.

[0028] the lock-mode information about said data information file corresponding to each lock information file and said schema information file -- further, The lock-mode information about the schema information (namely, information which shows the structure of the column information used as the attribute of the data record of said data information file) which is a component of the data record which is a component of said data information file and said schema information file is stored. A race condition is controlled and it has two or more lock modes corresponding to the classification (for example, search, read-out, updating, an addition, deletion, etc.) of a data-

processing demand as a lock mode for the prevention from competition which enables exclusive control processing.

[0029]The data information file in the database file 131 which constitutes database DB130In the case of the relational data base model (a related data base modelRelationalData Model) widely adopted as latest commercial database managerial system (DBMS). It comprises a tabular format and the database file 131 is expressed as a meeting of a table (table Table). In this casevertical row (field) (column Column)i.e.sequenceas which each table (table) specifies two or more attributes (attribute attribute)It is made into the componenthorizontal row (record) (low Row)i.e.linewhich specify the combination (tuple tuple) of each of this attribute (attribute). Each data structure of each schema information file which saves each data information file and this schema will be defined by specifying each table (table) and each sequence (column) here as schema (schema) information.

[0030]The database file 131the lock file 132and the schema file 133 are usually memorized on the hard disk 13as shown in drawing 2but. When the memory 12 has a margin as a storage capacityit is good also as making a part or all of this database file 131the lock file 132and the schema file 133 save on the memory 12.

[0031]Database management system DBMS120 [ next]Have the inquiry request reception means 124the inquiry optimizing means 125the inquiry execution means 126the inquiry result transmitting means 127and the access plane 128and furtherIt has the database file access means 121the lock file accessing means 122and the schema file accessing means 123. The inquiry request reception means 124 receives inquiry demand SQL (Structured Query Language) to the database from a terminalchecks the propriety of a receptionistand generates the transaction for a data-processing demand.

[0032]Based on the generated transaction for a data-processing demandthe inquiry optimizing means 125Referring to the various pertinent information which conducts syntax analysis and a semantic analysis about inquiry demand SQL to the received databaseand is in the schema file 133 or the lock file 132. The optimal access path to the database file 131 grade according to this inquiry demand is determinedand the access plane 128 with the shortest execution time is generated.

[0033]Herethe access plane 128 is recording here the internal processing code which showsthe execution sequencei.e.the data-processing procedureof data processing about the database access path by which optimization was attained by the inquiry optimizing means 125and shows the order of the real way of inquiry demand SQL. The access plane 128 is saved in principle on the memory 12 shown in drawing 2.

[0034]The inquiry execution means 126referring to the lock-mode information in the lock file 132 based on the access plane 128 generated by the inquiry optimizing means 125. The data information file of the request in the database file 131Or access to the schema information file of the request in the schema file 133 is performeddata processing based on the inquiry demand shown in the access plane 128 is

performed and the inquiry result as a data processing result is outputted. The inquiry result transmitting means 127 returns the inquiry result which the inquiry execution means 126 outputted to the terminal of a requiring agency via the communication network 40.

[0035] The database file access means 121 According to the data-processing demand classification from the inquiry execution means 126 the data information file of the request in the database file 131 is accessed and radial transfers such as search about this data information file read-out updating an addition or deletion is performed.

[0036] The lock file accessing means 122 The inquiry optimizing means 125 Or based on the demand from the inquiry execution means 126 the lock information file of the request in the lock file 132 is accessed and radial transfers such as search about this lock information file read-out updating an addition or deletion is performed.

[0037] The schema file accessing means 123 The inquiry optimizing means 125 Or based on the demand from the inquiry execution means 126 the schema information file of the request in the schema file 133 is accessed and radial transfers such as search about this schema information file read-out updating an addition or deletion is performed.

[0038] Next operation of the outline of database management system DBMS concerning this invention is explained based on the flow chart shown in drawing 5. First when the data-processing demand from a terminal asks and it is received by the request reception means 124 via the communication network 40 the inquiry request reception means 124 After performing acceptance processing of this data-processing demand the transaction for this data-processing demand is made to generate and the transaction for this data-processing demand is transmitted to the inquiry optimizing means 125 (S1).

[0039] The inquiry optimizing means 125 which received the transaction for this data-processing demand The data-processing procedure according to the data-processing demand from the inquiry reception means 124 included in this transaction is made to generate based on the schema information and the lock-mode information relevant to this data-processing demand and is added to the search space one by one (S2). Via the schema file accessing means 123 schema information is read from the schema information file of the request in the schema file 133 and lock-mode information It is read from the lock information file of the request in the lock file 132 via the lock file accessing means 122.

[0040] Out of the data-processing procedure in the generated search space the inquiry optimizing means 125 checks the shortest execution time and the possibility of parallel operation draws up the access plane 128 used as the optimal data-processing procedure and sends it out to the inquiry execution means 126 (S3). To the drawn-up access plane 128. [ in order to perform exclusive control for the prevention from competition of a data-processing demand before and after performing / of data processing ] Set up the lock mode about a data object (data record in the data

information file of a processing object or a data information file) or a schema object (schema information in the schema information file and schema information file of a processing object) or The data-processing procedure to cancel is also included.

[0041] Based on the drawn-up access plane 128 by the inquiry optimizing means 125 the inquiry execution means 126 via the lock file accessing means 122 check the race condition of the data object of a processing object or a schema object and (S4) When it is in a race condition the Reason of the purport that data processing which performed waiting processing or was demanded cannot be performed is attached until (YES of S4) and other data processing which is competing are completed and a rejected note is transmitted to the inquiry result transmitting means 127. In crawling 5 after waiting the timing of the time limit set beforehand (S10) in order to check a race condition again the processing in the case of returning to step S4 is shown.

[0042] On the other hand if it is in the state where execution of data processing of said data object or said schema object is permitted (NO of S4) this data object and this schema object are received Writing processing is performed to the specified lock information file in the lock file 132 via the lock file accessing means 122 in order to make the lock mode of the request corresponding to data-processing demand classification set up (S5).

[0043] Then the inquiry execution means 126 performs data processing described by the access plane 128 to said data object or said schema object of a processing object (S6). Namely when data processing described by the access plane 128 is pointing to search read-out Via the database file access means 121 or the schema file accessing means 123 the inside of the database file 131 or the schema file 133 is searched and said data object or said schema object is read. When the data-processing demand described by the access plane 128 is pointing to the data update process to said data object or said schema object After performing the data update process according to this data-processing demand to said data object or said schema object It writes in via the database file access means 121 or the schema file accessing means 123 into the database file 131 or the schema file 133.

[0044] When data processing based on the access plane 128 is completed in order to make the lock mode set as said data object or said schema object cancel Writing processing to the specified lock information file in the lock file 132 is performed via the lock file accessing means 122 (S7). The inquiry execution means 126 outputs the result of having performed the data processing means based on the access plane 128 to the inquiry result transmitting means 127 as an inquiry result (S8). The inquiry result transmitting means 127 transmits this inquiry result to the terminal of a requiring agency via the communication network 40 (S9).

[0045] Next in this invention the locking mechanism about the lock mode provided in competition prevention of the data-processing demand is explained. In the conventional database management system [ as are mentioned above and the

transaction of the data-processing demand is performed (that is data processing is performed) ] Let it be a premise not to update the schema information file which defines the structure of a data information file (that is data reconstruction processing should not be made). As the transaction of which [ including not only a data information file but a schema information file ] the data-processing demand is performed in this invention (that is data processing is performed) even if it is Not only a data information file but a schema information file is also received so that the change processing of a schema information file may become possible The completely same locking mechanism as a data information file is introduced and it is supposed that the change processing of a schema information file is also treated as a transaction for a kind of data-processing demand. By introducing this locking mechanism the data-processing demand to the usual data information file The parallel operation nature which can receive simultaneously the data-processing demand (a schema change processing demand i.e. a data reconstruction demand) to a schema information file and to which a concurrency can be carried out is secured.

[0046] As a lock mode for realizing the locking mechanism for the prevention from data-processing demand competition concerning this invention five lock modes X S I X S I X and I S are prepared to each for a data information file and for a schema information file. The compatibility (Compatibility) between these five lock modes is shown in drawing 3. The object same here as compatibility (that is) As opposed to the schema information which is a component in the data record which is a component in the data information file which is the target of a data-processing demand or a schema information file and a data information file or a schema information file the transaction of two or more data-processing demands exists and it performs simultaneously -- it is going to have -- \*\*\*\*\* [ that the lock mode which each transaction specifies as a case can be set up simultaneously ] (is it compatible or not?) -- the character to say is shown. In drawing 3 it is shown that "Y" can be compatible and "N" shows that it can be incompatible.

[0047] and also it receives the object to which it is exclusive mode (Exclusive Mode) in X and this exclusive mode X is set in drawing 3 -- from -- it is a lock mode which refuses a | data-processing demands (search reading updating an addition deletion etc.). It is a lock mode set up when performing data processing generally specified in the UPDATE statement which requires change of the object for data processing an addition and deletion the INSERT sentence the DELETE sentence etc.

[0048] moreover -- and also it performs reading operation to the object to which S is share mode (Shared Mode) and this share mode S is set -- from although it can be made to receive and perform in parallel about a data-processing demand It is a lock mode entirely refused about the data-processing demand of change an addition deletion etc. It is a lock mode set up when performing data processing generally specified by the SELECT statement etc. which require search of the object of a data-processing demand and reading processing. However even if it is a case of a

SELECT statement of course it is possible to also make the exclusive mode X carry out forcible setting out exceptionally.

[0049]IX is intent exclusive mode (Intent Exclusive Mode) and is a lock mode which shows that the exclusive mode X is set up to the component in the object to which this intent exclusive mode IX is set. Intent exclusive mode IX is what is set up to the table (table) which constitutes a data information file and a schema information file. When the exclusive mode X is set to the table (table) itself it is not necessary to make intent exclusive mode IX set up to this table (table).

[0050]IS is intent share mode (Intent Shared Mode) and is a lock mode which shows that share mode S is set up to the component in the object to which this intent share mode IS is set. Intent share mode IS is what is set up to the table (table) which constitutes a data information file and a schema information file. When share mode S is set to the table (table) itself it is not necessary to make intent share mode IS set up to this table (table).

[0051]SIX is share mode with intent exclusion (Shared with Intent Exclusive Mode). Although the table (table) object to which this share mode SIX with intent exclusion is set was set as share mode S and data processing was once under execution. Since data processing to which the exclusive mode X is made to set to a certain component in this table (table) object occurred it is a lock mode which shows that he wants to also make intent exclusive mode IX set as this table (table) object. For example when it is in the state where performed the SELECT statement which carries out repetitive reading operation to the big table (table) and share mode S is set up to this big table (table) by an UPDATE statement. When the demand which wants to update only one line (row) which is a component in this table (table) occurs and the exclusive mode X is set up to this 1 \*\* line (row) to this table (table) from share mode S the share mode SIX with intent exclusion will be made to change automatically and it will be reset as it.

[0052]That is intent exclusive mode IX, intent share mode IS and the share mode SIX with intent exclusion are lock modes provided in order to raise the concurrency nature of two or more data-processing demands to a data information file or a schema information file. For example when performing the data-processing demand which updates only the line (row) or sequence (column) which is a component in a specific table (table). When the concurrency of the reference request to other lines (row) or sequences (column) in this same table (table) is permitted, intent exclusive mode IX will be set up to this table (table) and the exclusive mode X will be set up only to the line (row) or sequence (column) used as an update object.

[0053]This lock mode will be set up like the above-mentioned also not only to a data information file but to a schema information file. Namely for example even if it is a case where the object for data processing is in a data information file, parallel operation of data processing to a data information file and data processing to a schema information file can be realized by making setting out of a lock mode simultaneously.

also to the schema information file not only relevant to a data information file but this data information file.

[0054] Next an easy data information file and a schema information file are taken for an example and also operation of the locking mechanism of the data-processing demand to a data information file or a schema information file is explained in detail. It is a figure showing an example of the schema information file i.e. the composition of a schema object which defines the data structure of this data object the data information file i.e. the data object in which drawing 4 is the target of a data-processing demand and here.

[0055] That is the sequence (column) C1 and Table R which defines the data structure about C2 exist as a schema information file in the schema file 133. Here suppose that it expresses schema object S (R) S (C1) and S (C2) corresponding to Table R the sequence (column) C1 and C2 respectively. On the other hand as a data information file in data \*\* - SUFAIRU 131 Table R exists as a data information file by which the data structure is defined by said schema object and Table R which is this data information file. The table (table) which becomes each line (low) of every from three lines (low)  $t_1$ ,  $t_2$  and  $t_3$  as the sequence (column) C1 and tuple (tuple) information that it has a data attribute (attribute) corresponding to C2 is constituted. Here suppose that it expresses data object D (R) D ( $t_1$ ) D ( $t_2$ ) and D ( $t_3$ ) corresponding to this table R line (low)  $t_1$ ,  $t_2$  and  $t_3$  respectively.

[0056] First the example of a locking mechanism of operation is explained only paying attention to the setting-operation of the lock mode about a data information file. For example when the data-processing demand transaction which searches the data of the data information file R i.e. a table one by one occurs a lock mode is set as the whole table R. When searching one applicable to the conditions of the data-processing demand concerning the whole table R of lines (low) in this case and following renewal of the data of this line (low) on it the exclusive mode X is set to whole table R D (R) i.e. a data object. Therefore all other data-processing demand transactions that require access to Table R until this retrieval processing is completed Since it will be in a race condition to the table R D (R) i.e. a data object as shown in drawing 3 it will be in the waiting state which returns to step S4 again via Step S10 through the judgment of YES of step S4 of drawing 4.

[0057] On the other hand it will be in the state where share mode S is set up to whole table R D (R) i.e. a data object only in the case of the read operation of a data file i.e. the data about the whole table R. In this state other data-processing demand transactions which require access to Table R in parallel In the data-processing demand which performs only read-out of the data of the specific line (low) in Table R In the time of execution of the data-processing demand transaction of these others Since IS mode will be set up to the whole table R and S mode will be set up to the line for read-out (low) as shown in drawing 3 the race condition over the table R D (R) i.e. a data object has not been generated. Therefore since the concurrency operation of both



data-processing demand transaction is attained through the judgment of NO of step S4 of drawing 4 in this case improvement in the speed of database processing can be attained.

[0058] When the data-processing demand transaction which performs renewal of data only to specific line (low)  $t_1$  of the data information file R i.e. a table is performed Intent exclusive mode IX is set up to whole table R D (R) i.e. a data object and the exclusive mode X is set up to line (low)  $t_1$  ( $t_1$ ) D of the object which performs renewal of data i.e. a data object. Therefore as shown in drawing 3 to line (low)  $t_1$  ( $t_1$ ) D i.e. a data object all other data-processing demand transactions will be in a race condition and will wait execution until said renewal of data is completed but. It is in the state where read and write [ both ] in to the data-processing demand transaction which requires access from other lines (low) D ( $t_2$ ) thru/or D ( $t_3$ ) i.e. data objects in Table R and concurrency operation is permitted and contributes to improvement in the speed of database processing.

[0059] When the data-processing demand transaction which performs a data information file i.e. the data read only to specific line  $t_1$  of Table R is performed Intent share mode IS is set up to whole table R D (R) i.e. a data object and share mode S is set up to line (low)  $t_1$  D for data read i.e. a data object ( $t_1$ ). Therefore until said data read of line (low)  $t_1$  is completed As shown in drawing 3 although data changing is required from line (low)  $t_1$  ( $t_1$ ) D i.e. a data object a data-processing demand transaction besides the time will be in a race condition and will wait but. About the read operation to line (low)  $t_1$  ( $t_1$ ) D i.e. a data object. With what kind of access operation to other lines (low) D ( $t_2$ ) thru/or D ( $t_3$ ) i.e. data objects of Table R the concurrency operation of other data-processing demand transactions is attained and they can expect improvement in the speed of database processing further.

[0060] In the explanation mentioned above although explained only paying attention to the setting-operation of the lock mode about a data information file When performing access operation to this data information file as mentioned above share mode S and/or intent share mode S are simultaneously set up also to the schema information file which defines the data structure of the data information file which is the target of data processing. Namely in performing the data-processing demand transaction to the whole table R which is the target data information file. Share mode S is set up to the schema object S (R) corresponding to the whole schema information file i.e. table R which defines the data structure of Table R which is this data information file. On the other hand in performing the data-processing demand transaction to the specific line (low) in said data information file R i.e. a table Are in the schema object S (R) and Line (low)  $t_1$   $t_2$  and  $t_3$  ( $t_1$ ) D i.e. the data object of said data information file. As opposed to the schema objects S (C1) and S (C2) which define the data structure i.e. the structure of the sequences (column) C1 and C2 of D ( $t_2$ ) and D ( $t_3$ ) Share mode S is set up and intent share mode IS is set up to the schema object S (R) which shows Table R.

[0061] Also as opposed to the schema information file corresponding to the data

information file this profit and for data processing. Even if it is in the state which the data-processing demand transaction to a data information file is performing by setting up a lock mode simultaneously. It becomes possible to make it operate in parallel and although the access processing to a database file is stopped during renewal of a schema information file it can make inconvenient [ of employment ] avoid conversely at the time without making the update process of a corresponding schema information file compete.

[0062] Next the setting-operation of the lock mode about a schema information file, i.e. when the data-processing demand transaction to a schema information file has occurred is explained. Concerning [ first ] the schema information file which has specified the data structure of Table R shown in above-mentioned drawing 4. When the data-processing demand transaction T1 which uses the definition content of the sequence (column) C1 as a change plug exists in the inquiry execution means 126 until the change processing of a definition content is completed. Intent exclusive mode IX is set up to the schema object S (R) corresponding to table R which the exclusive mode X is set up to the schema object S (C1) which is the target of change processing and has this schema object S (C1).

[0063] In the state where this data-processing demand transaction T1 precedes and exists. When the data-processing demand transaction T2 on which the contents defined as the sequences (column) C1 and C2 are displayed carried out late-coming and it generates. The inquiry execution means 126 which is going to perform this transaction T2. As opposed to the schema objects S (C1) and S (C2) which are the targets of data processing. Let intent share mode IS be a setting-out \*\* plug to the schema object S (R) corresponding to table R which uses share mode S as a setting-out \*\* plug and has these schema objects S (C1) and S (C2).

[0064] However, since the exclusive mode X set up to the schema object S (C1) does not accept coexistence to be any lock modes as shown in drawing 3 both transactions T1 and T2 will be in a race condition in the schema object S (C1).

[0065] Therefore the late-coming transaction T2 until it is judged with the race condition having arisen in step S4 of the flow chart shown in drawing 5 and the definition change processing of the sequence (column) C1 by the transaction T1 of a forerunner is completed. By repeating the loop which returns to step S4 via Step S10 of the flow chart shown in drawing 5 it is considered as a waiting state.

[0066] In the state where the aforementioned transaction T1 precedes and exists. In the case where transaction T3 for a data-processing demand on which only the contents defined as the sequence (column) C2 of Table R are displayed carried out late-coming and it generates. The inquiry execution means 126 which is going to perform this transaction T3. Let intent share mode IS be a setting-out \*\* plug to the schema object S (R) corresponding to table R which uses share mode S as a setting-out \*\* plug and has this schema object S (C2) to the schema object S (C2) which is the target of processing.

[0067] Since the race condition is not generated between both transactions T1 and T3 in this state as shown in drawing 3 In step S4 of the flow chart shown in drawing 5 it judges that late-coming transaction T3 is not a race condition either and the transaction T1 of a forerunner and the operation of a concurrency of it are attained. Therefore improvement in the speed of database processing can be attained by both concurrency.

[0068] When the transaction for a data-processing demand which makes the schema object which defines the data structure of the data information file R i.e. a table change like the aforementioned transaction T1 is performed It is not necessary to set up any lock modes to the data object D (R) side in which it is shown the data information file R i.e. the table which are defined by this schema object. However like the aforementioned transaction T2 or T3 In the case where a transaction which is displayed is performed without changing the schema object which defines the data structure of Table R After setting out of the lock mode to a schema object is completed by this schema object. Data object D (R) which shows the data information file R i.e. the table where the data structure is defined And a lock mode must be set up in order to prevent competition of data processing to the data object D (t<sub>1</sub>) D (t<sub>2</sub>) and D (t<sub>3</sub>) in which each line (low) which is a component of this table R is shown. Namely the aforementioned transaction T2 or T3 is performed Each schema object S (R) S (C1) S (C2) or S (R) As opposed to the data object D (t<sub>1</sub>) D (t<sub>2</sub>) and D (t<sub>3</sub>) in which each line (low) of Table R defined by this schema object is shown after setting out of the lock mode to S (C2) is completed Share mode S is set up and intent share mode IS is set up to the data object D (R) in which Table R is shown.

[0069] Therefore in the state where the aforementioned transaction T2 starts first and is processed for example The data object D in which a data structure is shown by the schema object S (C2) (t<sub>1</sub>). In the case where the data-processing demand transaction T4 which updates the data content of D (t<sub>2</sub>) and D (t<sub>3</sub>) carried out late-coming and it generates perform this transaction T4 — the inquiry execution means 126 which \*\*\*\* As opposed to the data object D (t<sub>1</sub>) D (t<sub>2</sub>) and D (t<sub>3</sub>) in which each line (low) of the data information file R i.e. a table defined by the schema object S (C2) is shown Use exclusive mode X as a setting-out \*\* plug and let intent exclusive mode IX be a setting-out \*\* plug to the data object D (R) in which Table R is shown. The schema object S (C2) corresponding to the sequence C2 in each line (low) of the data information file which is a processing object of said transaction T4 is also received Share mode S is made to set up and let intent share mode IS be a setting-out \*\* plug to the schema object S (R) corresponding to table R which has this schema object S (C2).

[0070] In this state if it sees about rivalry with both the transactions T2 and T4 to a schema object about the transaction T2 Share mode S receives the schema objects S (C1) and S (C2) again at the schema object S (R) Intent share mode IS is set up and on the other hand intent share mode IS tends to be set to the schema object S (C2) for

share mode S to the schema object S (R) about the transaction T4 again. As shown in drawing 3 among both transactions T2 and T4 it does not generate but concurrency operation is possible for rivalry.

[0071] However in [ if rivalry with the transactions T2 and T4 of both to a data object is seen ] the time of execution of the transaction T2 Share mode S is set as the data object D ( $t_1$ ) D ( $t_2$ ) and D ( $t_3$ ) and intent share mode IS is set as the data object D (R). On the other hand about the transaction T4 the data object D ( $t_1$ ) D ( $t_2$ ) and D ( $t_3$ ) is received the exclusive mode X -- intent exclusive mode IX sets up to the data object D (R) -- it is going to have -- a sake -- drawing 3 -- being shown -- as -- both -- the transactions T2 and T4 serve as rivalry. Therefore the late-coming transaction T4 will be made into a waiting state.

[0072] Next as schema information which has specified the structure of Table R shown in drawing 4 the index I is defined as the sequence (column) C1 and the case where Table R is connected with other tables by the index I is explained. In the case where the data-processing demand transaction T5 of schema information change which reconstructs this index I (remaking) occurs to schema OBUSHIEKUTO S(I) which shows the index I the exclusive mode X will be set up and intent exclusive mode IX will be set as schema object S (R) corresponding to Table R.

[0073] In the case where a data-processing demand transaction which searches with a SELECT statement etc. the data stored in the sequence (column) C1 of the data information file shown in Table R here occurs simultaneously Since it is not accompanied by change of the data stored in the sequence (column) C1 it is not necessary to schema object S(I) to set up a lock mode.

[0074] However in the case where a data-processing demand transaction which changes the data stored in the sequence (column) C1 of the data information file shown in Table R by an UPDATE statement etc. occurs Share mode S must be made to set up also to schema object S(I). Namely when the index I tends to be reconstructed as schema information When the data-processing demand transaction which uses as a change plug the data stored in the sequence (column) C1 of a data information file occurs By making generating of the competition to schema object S(I) detect and making waiting processing of the data changing of the sequence (column) C1 which is index data perform Parallel coincidence of both transaction can be made possible and generating of the situation which must perform data changing of the sequence (column) C1 of a data information file again after reconstruction of the index I can be prevented.

[0075] In the database management method and database management device which are applied to this invention as stated above Also not only to a data object but to a schema object by making the lock mode for the prevention from competition set up the parallel operation nature of data processing about a database was maintained and serialization possibility is guaranteed. That is when the data-processing demand transaction which changes schema information occurs for example a lock mode is set

up to the schema object used as a change target. In this case when a change target is a sequence (column) which specifies the attribute of data to the table (table) where this sequence (column) serves as a component in the exclusive mode X again intent exclusive mode IX is set up to this sequence (column). When a change target is concerned with front [ whole ] the exclusive mode X is set up to front [ used as a change target / whole ] but to each sequence (column) which is a component of this table it is not necessary to set up the exclusive mode X. When you need change also to the schema object which shows an index when the index is defined it sets up the exclusive mode X. When it is a data-processing demand transaction which changes schema information to the data object in which the data structure is defined it is not necessary to set up a lock mode by this schema information.

[0076] On the other hand when the data-processing demand transaction of only reference occurs without changing schema information a lock mode is set up to the schema object used as the candidate for reference. In this case when the candidate for reference is the sequence (column) or index which specifies the attribute of data to the table (table) where this sequence (column) or the index serves as a component in share mode S again intent share mode IS is set up to this sequence (column) or an index. It is necessary to set up a lock mode by this schema information also to the data object in which the data structure is defined. That is when setting out of the lock mode to a schema object is completed a lock mode is set up also to a corresponding data object. Here share mode S is set up to each line (row) which serves as a component of the table of this data object in intent share mode IS or share mode S again to the table (table) which is this data object.

[0077] Namely in the database management method and database management device concerning this invention Data processing by the data definition language (DDL: Data Definition Language) which makes the schema information which defines the structure of a data information file generate as a schema information file Namely data processings such as search read-out change an addition deletion etc. to a schema object data processing by the data manipulation language (DML: Data Management Language) which performs data manipulation to a data information file provided with the data substance by which the data structure is defined by the schema information in this schema information file — that is In both sides with data processings such as search read-out change an addition deletion etc. to a data object Serialization possibility is also guaranteed while securing the parallel operation nature of this data processing by making data processing perform setting up the lock mode to the both sides of a schema object and a data object.

[0078] With serialization possibility about two or more data-processing demand transactions here The state of the database which made it parallel made perform data processing and was obtained shows the character of data processing which is in agreement with the state of the transaction which is in agreement with the state of the database produced by making perform these two or more data-processing

demand transactions one by one according to a certain order. If this serialization possibility is guaranteed even if it makes it perform by making parallel two or more data-processing demand transactions to the database which has a certain consistency it can guarantee making the state of a database of having new consistency change.

[0079] The contents of the access plane 128 which shows the optimal procedure performed by the inquiry execution means 126 can also be made to change in the inquiry optimizing means 125 concerning this invention based on the state of the lock mode set up to the schema object. [ for example/ when the inquiry optimizing means 125 tends to generate the access plane 128 according to a new data-processing demand ] Since it also has the lock mode to the schema information about this index information change when it is a situation where the data-processing demand transaction of schema information change which reconstructs the index relation of a database is performed It is possible to also make the access plane 128 which does not use the data related to reconstruction index-related [ said ] generate in consideration of the parallel operation nature of data processing.

[0080] Corresponding to the data-processing demand (data-processing demand of searchread-outchangeadditiondeletionetc. to a data object or a schema object) based on the trigger and assertion function which are functions made to extend further an inquiry function only not only in a data object By making the lock mode to a related schema object set up By the trigger the schema object relevant to an assertion function or this schema object. Parallel operation nature with the data-processing demand (data-processing demand of searchread-outchangeadditiondeletionetc.) transaction to the data object defined can also be realized. When the specific conditions beforehand determined as the trigger or the assertion function here are satisfied It is the function it was made to start automatically the data-processing procedure defined beforehand for example when some salary data is changed it is a function which starts the operation which computes the average data of a salary of all the members automatically and is made to reset up.

[0081]

[Effect of the Invention] By providing the lock mode to the schema information which defines the data structure of each data information file which constitutes a database Even if it is in the middle of execution of the transaction about the demand of data processing (namely in the middle of [ execution / of data processing ]) Since it is possible to make a change of the schema information about the data information file concerning this data processing make in parallel and the number of data-processing demands in which a concurrency is possible can be made to increase the rapidity of the whole database management system and high processing efficiency can be made to attain.

[0082] Since the suspension of a database management system can be made to avoid at the time of change of schema information simplification can be attained for the

operation procedure of a database management system and operation cost can be reduced.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1] It is a block lineblock diagram showing various kinds of processings which constitute database management system DBMS concerning this invention and various kinds of information files which constitute the database DB used as the processing object of this processing.

[Drawing 2] It is a block lineblock diagram showing the hardware constitutions of the database system concerning one working example of this invention.

[Drawing 3] It is a matrix chart showing the compatibility of the lock mode which realizes the locking mechanism concerning this invention.

[Drawing 4] It is a figure showing an example of the composition of the data object which is the target of a data-processing demand and a schema object.

[Drawing 5] It is a flow chart for explaining operation of the outline of database management system DBMS concerning this invention.

[Description of Notations]

10 -- A server 112131 -- CPU 122232 -- Memory 132333 -- A hard disk 142434 -- Keyboard 152535 -- A pointing device 162636 -- Display 2030 -- The terminal 1 the terminal N 40 -- A communication network 120 -- Database management system DBMS 121 -- A database file access means 122 -- Lock file accessing means 123 -- A schema file accessing means 124 -- Inquiry request reception means 125 -- An inquiry optimizing means 126 -- An inquiry execution means 127 -- Inquiry result transmitting means 128 [ -- A lock file 133 / -- A schema file 221321 / -- An inquiry request input means 222322 / -- Inquiry result display means. ] -- An access plane 130 -- The database DB 131 -- A database file 132

---

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号  
特開2002-149450  
(P2002-149450A)

(43)公開日 平成14年5月24日(2002.5.24)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード*(参考)
G 0 6 F 12/00	5 0 1	G 0 6 F 12/00	5 0 1 B 5 B 0 7 5
	5 3 5		5 3 5 Z 5 B 0 8 2
17/30	1 8 0	17/30	1 8 0 D
	2 4 0		2 4 0 A

審査請求 未請求 請求項の数8 O L (全 16 頁)

(21)出願番号 特願2000-348963(P2000-348963)

(22)出願日 平成12年11月16日(2000.11.16)

(71)出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72)発明者 浅田 一繁

東京都大田区中馬込1丁目3番6号 株式  
会社リコー内

(72)発明者 竹川 弘志

東京都大田区中馬込1丁目3番6号 株式  
会社リコー内

(74)代理人 100079843

弁理士 高野 明近 (外2名)

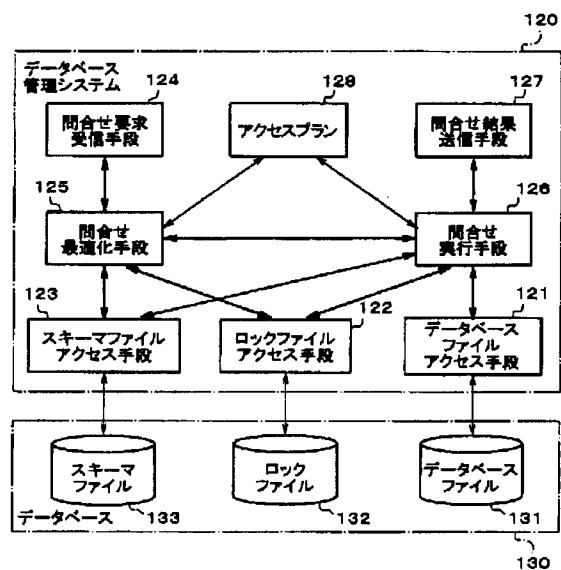
最終頁に続く

(54)【発明の名称】 データベース管理方法及びデータベース管理装置

(57)【要約】

【課題】 データベースのデータ処理を実行中でも、スキーマ情報の変更を可能とする。

【解決手段】 複数のデータ情報ファイルからなるデータベースファイル131と、複数のスキーマ情報ファイルからなるスキーマファイル133と、前記データ情報ファイル及び前記スキーマ情報ファイル毎に、それぞれデータ処理要求種別に対応したロックモードを格納することができるロックファイル132と、新たなデータ処理要求の発生の都度、ロックファイル132に格納されている前記データ情報ファイル及び前記スキーマ情報ファイル毎のロックモードを参照して、発生したデータ処理要求の競合状態をチェックし、最適のデータ処理の実行手順を生成する問合せ最適化手段125と、該実行手順に従って、要求されたデータ処理の実行に先立ち、ロックモードをロックファイル132に設定し、前記データ処理の終了時点で、設定したロックモードを解除させる問合せ実行手段126とを有している。





**【特許請求の範囲】**

【請求項1】 複数のデータ情報ファイルからなるデータベースファイルと、該データ情報ファイルのそれぞれのデータ構造を定義する複数のスキーマ情報ファイルからなるスキーマファイルとを有し、前記データ情報ファイル及び前記スキーマ情報ファイルに対するデータ処理要求の競合を防止するための複数のロックモードを前記データ処理要求の種別に応じて設け、該ロックモードを前記データ情報ファイル毎及び前記スキーマ情報ファイル毎に前記データ処理要求の種別に対応して設定させることにより、前記データ情報ファイルに対する複数のデータ処理の並行動作のみならず、前記データ情報ファイルと前記スキーマ情報ファイルとに対する複数のデータ処理の並行動作及び前記スキーマ情報ファイルに対する複数のデータ処理の並行動作を可能とすることを特徴とするデータベース管理方法。

【請求項2】 請求項1に記載のデータベース管理方法において、前記データ情報ファイルと前記スキーマ情報ファイルとが、それぞれ表形式のファイル形式により構成されており、前記ロックモードの設定を、前記データ情報ファイル及び前記スキーマ情報ファイルを構成するそれぞれの表単位に、前記データ処理要求の種別に対応して設定させることにより、表を構成する前記データ情報ファイルに対する複数のデータ処理の並行動作のみならず、表を構成する前記データ情報ファイルと前記スキーマ情報ファイルとに対する複数のデータ処理の並行動作及び表を構成する前記スキーマ情報ファイルに対する複数のデータ処理の並行動作を可能とすることを特徴とするデータベース管理方法。

【請求項3】 請求項2に記載のデータベース管理方法において、表形式の前記データ情報ファイル及び前記スキーマ情報ファイルのそれぞれのファイルの構成要素となるデータレコードである行単位毎及びデータ属性である列単位毎に、前記ロックモードを前記データ処理要求の種別に対応して設定させることにより、前記データ情報ファイルに対する複数のデータ処理の並行動作のみならず、前記データ情報ファイルと前記スキーマ情報ファイルとに対する複数のデータ処理の並行動作及び前記スキーマ情報ファイルに対する複数のデータ処理の並行動作を可能とすることを特徴とするデータベース管理方法。

【請求項4】 請求項1乃至3のいずれかに記載のデータベース管理方法において、前記データ情報ファイルが、複数の前記データ情報ファイル間の関連付けを行なう索引データを有し、前記スキーマ情報ファイルが、該索引データの索引関係を定義する索引情報を有している場合において、前記データ情報ファイル内の前記索引データ毎及び前記スキーマ情報ファイル内の前記索引情報毎に、前記ロックモードを前記データ処理要求の種別に対応して設定させることにより、前記スキーマ情報ファ

イル内の前記索引情報に対する1乃至複数のデータ処理と、該索引情報が定義している前記データ情報ファイル内の対応する前記索引データに対する1乃至複数のデータ処理との並行動作を可能とすることを特徴とするデータベース管理方法。

【請求項5】 請求項1乃至4のいずれかに記載のデータベース管理方法において、あらかじめ定められた条件が成立した場合にあらかじめ定められたデータ処理を自動的に実行させるトリガ及びアサーション機能を有している場合に、該トリガ及びアサーション機能によるデータ処理要求に対応して、前記データ情報ファイル及び前記スキーマ情報ファイルに前記ロックモードを設定させることにより、前記トリガ及びアサーション機能に基づく前記データ情報ファイルあるいは前記スキーマ情報ファイルに対する1乃至複数のデータ処理と、前記トリガ及びアサーション機能以外の他のデータ処理要求に基づく前記データ情報ファイルあるいは前記スキーマ情報ファイルに対する1乃至複数のデータ処理との並行動作を可能とすることを特徴とするデータベース管理方法。

【請求項6】 請求項1乃至5のいずれかに記載のデータベース管理方法において、前記データ情報ファイルあるいは前記スキーマ情報ファイルに設定されているロックモードの設定状況に基づいて、前記データ情報ファイルあるいは前記スキーマ情報ファイルに対するデータ処理の実行時間が最短となるように、該データ処理の実行手順を最適化させることを特徴とするデータベース管理方法。

【請求項7】 複数のデータ情報ファイルからなるデータベースファイルと、該データ情報ファイルのそれぞれのデータ構造を定義する複数のスキーマ情報ファイルからなるスキーマファイルとを有するデータベース管理装置において、前記データ情報ファイル及び前記スキーマ情報ファイルに対するデータ処理要求の競合を防止するための複数のロックモードを、前記データ処理要求の種別に応じて設け、前記データ処理要求の種別に対応して設定する前記ロックモードを前記データ情報ファイル毎及び前記スキーマ情報ファイル毎に格納させているロックファイルと、前記データ情報ファイルあるいは前記スキーマ情報ファイルに対するデータ処理要求が発生する都度、前記スキーマ情報ファイルに格納されているスキーマ情報及び前記ロックファイルに格納されている前記ロックモードとを参照して、発生した前記データ処理要求の競合状態をチェックし、最適のデータ処理の実行手順を生成する問合せ最適化手段と、生成された該実行手順に従って、要求されたデータ処理を実行させるとともに、該データ処理の実行に先立って、発生した前記データ処理要求に対応する前記ロックモードを前記ロックファイルに格納させ、前記データ処理の実行が終了した時点で、前記ロックファイルに格納させた前記ロックモードを解除させる問合せ実行手段とを有していることを特

徴とするデータベース管理装置。

【請求項8】 複数のデータ情報ファイルからなるデータベースファイルと、該データ情報ファイルのそれぞれのデータ構造を定義する複数のスキーマ情報ファイルからなるスキーマファイルとを有し、前記データ情報ファイル及び前記スキーマ情報ファイルに対するデータ処理要求の競合を防止するための複数のロックモードを前記データ処理要求の種別に応じて設け、該ロックモードを前記データ情報ファイル毎及び前記スキーマ情報ファイル毎に前記データ処理要求の種別に対応して設定させることにより、前記データ情報ファイルに対する複数のデータ処理の並行動作のみならず、前記データ情報ファイルと前記スキーマ情報ファイルとに対する複数のデータ処理の並行動作及び前記スキーマ情報ファイルに対する複数のデータ処理の並行動作を可能とするデータベース管理方法を、コンピュータに実施させるためのプログラムとして記録させているコンピュータ読取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、データベース管理システム（DBMS）、特に、リレーショナルデータベース管理システム（RDBMS）において、予備のデータベースを設置することなく、データベースサービスの運用を継続させながら、データベースを構成するデータ情報ファイルのデータ構造の変更（すなわちスキーマ情報の変更）を含むデータベースの再構成を可能とするデータベース管理方法に関する。また、かかるデータベース管理方法を実現させるデータベース管理装置、および、かかるデータベース管理方法をコンピュータにて実行させるプログラムとして記録させたコンピュータに読取り可能なプログラム記録媒体に関する。

【0002】

【従来の技術】 従来より、オンライン処理システムに適用されるデータベース管理システムにおいて、オンラインサービスを中断させずに、データベースを再構成させるためのいくつかの技術が開示されてきている。ここに、データベースの再構成とは、旧データベースの構造を変更して新しいデータベースの構造に変更させるものである。例えば、郵便番号の桁数が3桁から7桁に変更されたことに伴い、住所録データ情報ファイル内の郵便番号を記憶している3桁のデータ領域の大きさを変更して、新たな7桁のデータ領域に拡張させるとともに、拡張させたデータ領域に新たな7桁の郵便番号を格納させた住所録データ情報ファイルを作成させるものである。

【0003】 かかるデータベースの再構成をオンラインサービスを中断させずに実現させるものとして、例えば、特開平5-233408号公報「オンライン処理システムにおける移行方法」において開示されているデータベース再構成技術がある。該公報においては、二重化

構成のデータファイル記憶装置を用いて、片系でオンラインサービスを継続させたまま、もう一方の別系上で新たな構成のデータ情報ファイルへの移行処理を行い、移行処理完了時に、一旦、オンラインサービスを停止させて、新たな構成のデータ情報ファイルへ移行した別系にオンライン系を切り替え、新たな系でオンラインサービスを再開させることにより、オンラインサービスの中断時間を最小限に抑えようとするものである。

【0004】 更に、かかる二重化構成のデータファイル記憶装置を用いて、オンライン系への切り替え時に、オンラインサービスを停止させずに、二重化同期運転を実行させて、瞬時に切り替えを実現させて、オンラインサービス無中断のデータベース再構成技術も提案されている。

【0005】 しかしながら、このようなデータベースシステムは、二重化構成が前提となるため、定常状態においても、全く同様のデータ情報ファイルやプログラムファイルを収納する2つの記憶装置を必要とする。更には、二重化構成を制御するための特殊な各種制御機能も必要となり、高価なシステム構成となってしまう。すなわち、通常の単一のデータファイル記憶装置からなるデータベースシステムでは、実現させることができない。

【0006】 一方、予備系を有しない単一系のデータベースシステムにおいても、オンラインサービスの中断なく、データベースの再構成を実現させる技術としては、特開平7-244603号公報「データベース再構成方法」にて開示されているものがある。

【0007】 該公報において開示されている方法は、データファイル記憶装置上のデータ情報ファイルへのアクセス手段を物理的なデータファイル記憶装置上の場所や大きさを意識せずにアクセスする抽象データアクセス手段と、意識してアクセスする実データアクセス手段との2つに分離させ、実データアクセス手段がデータ構造の再構成を行った後においては、抽象データアクセス手段として、新旧間のデータ情報ファイルの変換手段を有せしめることにより、オンラインシステムの稼働状態のままで、新旧いずれのデータ情報ファイルに対してでも、データ情報ファイルの更新を行なうことを可能にせんとするものである。

【0008】 また、特開平7-325744号公報「データベース形式変換方法及び装置」においても、予備系を有しない単一のデータファイル記憶装置におけるオンラインサービス無中断データベース再構成を可能としている。該公報において開示されている方法は、データ情報ファイルのデータ構造の変換処理の要求が発出された時点で、元のデータ情報ファイルの更新は一旦凍結させ、該データ情報ファイルに対して施すべき更新記録を別に設けられたログファイルに逐次記録させていき、一方、元のデータ情報ファイルのデータ構造を、所望のデータ構造に換装させた後、新たなデータ形式に変換させ

ながら旧データ情報ファイルから順次コピーしていくものである。すべてのデータ構造の変換が終了した時点で、ログファイルに記録されているデータについて、新たなデータ形式に変換して作成し直して、更に上書きコピーしていく。

【0009】また、特開平08-077050号公報「データベース再構成装置及びデータベース再構成方法」においても、再構成対象の旧テーブルと旧テーブルの更新履歴情報を蓄積させるログファイルとを有し、旧テーブルの全ての情報を新テーブルに置換してコピーした後、ログファイルに蓄積されている更新履歴情報に基づいて、更新後の旧テーブルの情報について新テーブルに反映させていく無中断再構成方法を提案している。

【0010】

【発明が解決しようとする課題】しかしながら、従来の技術においては、予備系を有しない単一のデータベースシステムでは、データベースの実体データを有しているデータ情報ファイルに対するデータ処理要求を受け付けて、データ情報ファイルを更新させながら、同時に、データ情報ファイルの構造を変更させるがごときデータ再構成を行なわしめることはできず、該データ情報ファイルの構造の変更処理（すなわちデータ再構成処理）が終了するまで、受け付けたデータ処理要求にかかるデータ情報ファイルのデータ処理（データ更新処理）に関する情報を、例えば、別途用意させているログファイル等に保存させていく必要がある。

【0011】従って、同時にデータ処理要求が大量に発生し、更新された最新データを利用してオンライン処理を迅速に進める必要があるオンラインシステムにおいては、データベース処理の運用を一旦中断させる事に伴い、許容限界近傍まで遅延させた状態でオンライン動作をせざるを得ない状況が生じたり、オンライン動作を継続させたとしても、大量のログファイルが必要となるばかりか、最悪の場合には、オンライン動作に必要とするデータ情報ファイルに関する最新データを許容遅延時間内に取得できなくなり、オンライン動作自体も停止してしまう事態が生じかねない。

【0012】本発明は、かかる問題に鑑みてなされたものであり、データ情報ファイルに関する複数のデータ処理要求に対する競合防止用のロックモード以外に、該データ情報ファイルのデータ構造を定義するスキーマ情報ファイルに関する複数のデータ処理要求に対しても競合防止用のロックモードを設けることにより、データ情報ファイルに対する複数のデータ処理と、データ情報ファイルの構造を変更させる複数のデータ処理（すなわちデータ再構成処理）とを並行させて同時動作させることを可能にせんとするものである。

【0013】

【課題を解決するための手段】請求項1の発明は、複数のデータ情報ファイルからなるデータベースファイル

と、該データ情報ファイルのそれぞれのデータ構造を定義する複数のスキーマ情報ファイルからなるスキーマファイルとを有し、前記データ情報ファイル及び前記スキーマ情報ファイルに対するデータ処理要求の競合を防止するための複数のロックモードを前記データ処理要求の種別に応じて設け、該ロックモードを前記データ情報ファイル毎及び前記スキーマ情報ファイル毎に前記データ処理要求の種別に対応して設定させることにより、前記データ情報ファイルに対する複数のデータ処理の並行動作のみならず、前記データ情報ファイルと前記スキーマ情報ファイルとに対する複数のデータ処理の並行動作及び前記スキーマ情報ファイルに対する複数のデータ処理の並行動作を可能とすることを特徴としたものである。

【0014】請求項2の発明は、請求項1に記載のデータベース管理方法において、前記データ情報ファイルと前記スキーマ情報ファイルとが、それぞれ表形式のファイル形式により構成されており、前記ロックモードの設定を、前記データ情報ファイル及び前記スキーマ情報ファイルを構成するそれぞれの表単位に、前記データ処理要求の種別に対応して設定させることにより、表を構成する前記データ情報ファイルに対する複数のデータ処理の並行動作のみならず、表を構成する前記データ情報ファイルと前記スキーマ情報ファイルとに対する複数のデータ処理の並行動作及び表を構成する前記スキーマ情報ファイルに対する複数のデータ処理の並行動作を可能とすることを特徴としたものである。

【0015】請求項3の発明は、請求項2に記載のデータベース管理方法において、表形式の前記データ情報ファイル及び前記スキーマ情報ファイルのそれぞれのファイルの構成要素となるデータレコードである行単位毎及びデータ属性である列単位毎に、前記ロックモードを前記データ処理要求の種別に対応して設定させることにより、前記データ情報ファイルに対する複数のデータ処理の並行動作のみならず、前記データ情報ファイルと前記スキーマ情報ファイルとに対する複数のデータ処理の並行動作及び前記スキーマ情報ファイルに対する複数のデータ処理の並行動作を可能とすることを特徴としたものである。

【0016】請求項4の発明は、請求項1乃至3のいずれかに記載のデータベース管理方法において、前記データ情報ファイルが、複数の前記データ情報ファイル間の関連付けを行なう索引データを有し、前記スキーマ情報ファイルが、該索引データの索引関係を定義する索引情報を有している場合において、前記データ情報ファイル内の前記索引データ毎及び前記スキーマ情報ファイル内の前記索引情報毎に、前記ロックモードを前記データ処理要求の種別に対応して設定させることにより、前記スキーマ情報ファイル内の前記索引情報に対する1乃至複数のデータ処理と、該索引情報が定義している前記データ情報ファイル内の対応する前記索引データに対する1

乃至複数のデータ処理との並行動作を可能とすることを特徴としたものである。

【0017】請求項5の発明は、請求項1乃至4のいずれかに記載のデータベース管理方法において、あらかじめ定められた条件が成立した場合にあらかじめ定められたデータ処理を自動的に実行させるトリガ及びアサーション機能を有している場合に、該トリガ及びアサーション機能によるデータ処理要求に対応して、前記データ情報ファイル及び前記スキーマ情報ファイルに前記ロックモードを設定させることにより、前記トリガ及びアサーション機能に基づく前記データ情報ファイルあるいは前記スキーマ情報ファイルに対する1乃至複数のデータ処理と、前記トリガ及びアサーション機能以外の他のデータ処理要求に基づく前記データ情報ファイルあるいは前記スキーマ情報ファイルに対する1乃至複数のデータ処理との並行動作を可能とすることを特徴としたものである。

【0018】請求項6の発明は、請求項1乃至5のいずれかに記載のデータベース管理方法において、前記データ情報ファイルあるいは前記スキーマ情報ファイルに設定されているロックモードの設定状況に基づいて、前記データ情報ファイルあるいは前記スキーマ情報ファイルに対するデータ処理の実行時間が最短となるように、該データ処理の実行手順を最適化させることを特徴としたものである。

【0019】請求項7の発明は、複数のデータ情報ファイルからなるデータベースファイルと、該データ情報ファイルのそれぞれのデータ構造を定義する複数のスキーマ情報ファイルからなるスキーマファイルとを有する前記データ情報ファイル及び前記スキーマ情報ファイルに対するデータ処理要求の競合を防止するための複数のロックモードを、前記データ処理要求の種別に応じて設け、前記データ処理要求の種別に対応して設定する前記ロックモードを前記データ情報ファイル毎及び前記スキーマ情報ファイル毎に格納させているロックファイルと、前記データ情報ファイルあるいは前記スキーマ情報ファイルに対するデータ処理要求が発生する都度、前記スキーマ情報ファイルに格納されているスキーマ情報及び前記ロックファイルに格納されている前記ロックモードを参照して、発生した前記データ処理要求の競合状態をチェックし、最適のデータ処理の実行手順を生成する問合せ最適化手段と、生成された該前記に従って、要求されたデータ処理を実行させるとともに該データ処理の実行に先立って、発生した前記データ処理要求に対応する前記ロックモードを前記ロックファイルに格納させ、前記データ処理の実行が終了した時点で、前記ロックファイルに格納させた前記ロックモードを解除させる問合せ実行手段とを有していることを特徴としたものである。

【0020】請求項8の発明は、複数のデータ情報ファ

イルからなるデータベースファイルと、該データ情報ファイルのそれぞれのデータ構造を定義する複数のスキーマ情報ファイルからなるスキーマファイルとを有し、前記データ情報ファイル及び前記スキーマ情報ファイルに対するデータ処理要求の競合を防止するための複数のロックモードを前記データ処理要求の種別に応じて設け、該ロックモードを前記データ情報ファイル毎及び前記スキーマ情報ファイル毎に前記データ処理要求の種別に対応して設定させることにより、前記データ情報ファイルに対する複数のデータ処理の並行動作のみならず、前記データ情報ファイルと前記スキーマ情報ファイルとに対する複数のデータ処理の並行動作及び前記スキーマ情報ファイルに対する複数のデータ処理の並行動作を可能とするデータベース管理方法を、コンピュータに実施させるためのプログラムとして記録させていることを特徴としたものである。

【0021】

【発明の実施の形態】本発明の実施形態について、以下に、図を用いて説明する。図2は、本発明の一実施例にかかるデータベースシステムのハードウェア構成を示すブロック構成図である。図2においては、通信ネットワーク40を介して、端末120、…、端末N30と、データベースDB130を構築しているサーバ10とが相互接続されている。ここで、データベースDB130を利用するユーザが端末120、…、端末N30から所望のデータ情報ファイルに対する問合せ要求（すなわちデータ処理要求）を、通信ネットワーク40を介して送信すると、サーバ10は、データベースDB130を検索して、要求されたデータに対する処理を施して、そのデータ処理結果を問合せ要求を送信した端末に返送する。

【0022】ここに、端末120、…、端末N30は、PC等の計算機により構成されている。すなわち、各端末は、それぞれ、CPU21、…、31、メモリ22、…、32、ハードディスク23、…、33を備えており、各メモリ22、…、32には、それぞれ、サーバ10上のデータベースDB130に対する問合せ要求を発するための問合せ要求入力手段221、…、321と、問合せ結果を受け取り、表示するための問合せ結果表示手段222、…、322などを格納し、各CPU21、…、31により、プログラムとして実行させることができる。また、各端末には、それぞれ、データベースDB130への問合せ要求を入力するためのキーボード24、…、34及びポインティングデバイス（例えば、マウス）25、…、35が装備されており、また、データベースDB130へのそれぞれの問合せ結果を表示するためのディスプレイ26、…、36が装備されている。

【0023】また、通信ネットワーク40は、端末120、…、端末N30とサーバ10とを相互接続する

ための伝送路を構成するものであり、一般的には、有線ケーブルで実現されていて、TCP/IP等の通信プロトコルにより、端末120、…、端末N30とサーバ10間との情報交換がなされている。しかしながら、端末120、…、端末N30とサーバ10との通信プロトコルを一致させることができるならば、有線ケーブルに限らず、無線LAN用の電波や放送波を用いた無線ネットワークにより通信ネットワーク40を構築しても構わない。

【0024】また、サーバ10は、通信ネットワーク40を介して、端末120、…、端末N30から入力されたデータベースDB130への問合せ要求SQL

(Structured Query Language) すなわちデータ処理要求を受け取り、該問合せ要求に応じたデータ処理を施して、問合せ結果を該問合せ要求を発した端末120、…、端末N30に、通信ネットワーク40を介して、返送するための計算機である。すなわち、サーバ10は、マンマシンインタフェースを司るための入出力手段を提供するためのキーボード14、ポインティングデバイス15、あるいは、ディスプレイ16等を備えるとともに、CPRI11、メモリ12、ハードディスク13を備えていて、ハードディスク13上に格納されているデータベースDB130に対する各種の管理機能を司るデータベース管理システムDBMS120をメモリ12上のプログラムとして実行させることが可能な蓄積プログラム制御システムからなっている。

【0025】次に、サーバ10のメモリ12上に搭載されているデータベース管理システムDBMS120と、ハードディスク13に格納されているデータベースDB130の詳細なブロック構成を図1を用いて説明する。ここに、図1は、本発明にかかるデータベース管理システムDBMS120を構成する各種の処理(手段)と、該処理(手段)の処理対象となるデータベースDB130を構成する各種の情報ファイルとを示すブロック構成図である。データベースDB130はデータベースファイル131、ロックファイル132及びスキーマファイル133からなっている。データベースファイル131は、データベースの実体となるデータオブジェクトを、スキーマ(schema)として別途定義されるデータ構造を有するファイル形式(例えば、表形式)で蓄積している複数のデータ情報ファイルからなる情報ファイル群である。

【0026】ここに、スキーマ(schema)とは、各データ情報ファイルそのもの及び該データ情報ファイル中の各データの構造、形式、関連、制約等に関して記述したものであり、データ情報ファイルそのものあるいは該データ情報ファイルの個々の構成要素である各データの構造(以下、データ情報ファイルのデータ構造と略記する)を定義するものである。かかるスキーマに関する情報は、所定のファイル形式(例えば、表形式)で、スキ

ーマ情報ファイルとして、データ情報ファイルに対応させて作成される。作成された複数のスキーマ情報ファイルは、データベース130内にスキーマファイル133として蓄積されている。また、かかるスキーマ情報ファイル自体の、あるいは、スキーマ情報ファイルの個々の構成要素となるスキーマ情報の構造(以下、スキーマ情報ファイルのデータ構造と略記する)も、スキーマ情報ファイルとして保存している。

【0027】また、ロックファイル132は、データベースファイル131内のデータ情報ファイル及びスキーマファイル133内のスキーマ情報ファイルへのデータ処理要求の競合状態を制御する排他制御処理を可能とするためのロックモード情報を格納する情報ファイルであり、データベースファイル131内のデータ情報ファイル及びスキーマファイル133内のスキーマ情報ファイルそれぞれに対応して設けられた複数のロック情報ファイルから構成されている。

【0028】各ロック情報ファイルには、対応する前記データ情報ファイル及び前記スキーマ情報ファイルに関するロックモード情報、更には、前記データ情報ファイルの構成要素であるデータレコード及び前記スキーマ情報ファイルの構成要素であるスキーマ情報(すなわち、前記データ情報ファイルのデータレコードの属性となる列情報の構造を示す情報)に関するロックモード情報を格納している。なお、競合状態を制御し、排他制御処理を可能とする競合防止用のロックモードとしては、データ処理要求の種別(例えば、検索、読み出し、更新、追加、削除など)に対応して、複数のロックモードを有している。

【0029】また、データベースDB130を構成するデータベースファイル131内のデータ情報ファイルは、最近の商用データベース管理システム(DBMS)として広く採用されているリレーショナルデータベースモデル(関係データベースモデル、Relational Data Model)の場合、表形式で構成されており、データベースファイル131は、表(テーブル Table)の集まりとして表現される。この場合、各表(テーブル)は、複数の属性(アトリビュート attribute)を規定する縦の並び(フィールド)すなわち列(カラム Column)と、各該属性(アトリビュート)の組合せ(タプル tuple)を規定する横の並び(レコード)すなわち行(ロー Row)とを構成要素としている。ここに、各表(テーブル)と各列(カラム)とが、スキーマ(schema)情報として規定されることにより、各データ情報ファイル及び該スキーマを保存する各スキーマ情報ファイルのそれぞれのデータ構造が定義されることになる。

【0030】また、データベースファイル131、ロックファイル132及びスキーマファイル133は、通常、図2に示すように、ハードディスク13上に記憶されているが、メモリ12に記憶容量としての余裕がある

場合においては、かかるデータベースファイル131、ロックファイル132及びスキーマファイル133の一部あるいは全部をメモリ12上に保存させることとしても良い。

【0031】次にデータベース管理システムDBMS120は、問合せ要求受信手段124、問合せ最適化手段125、問合せ実行手段126、問合せ結果送信手段127、及び、アクセスプラン128を有し、更には、データベースファイルアクセス手段121、ロックファイルアクセス手段122、及び、スキーマファイルアクセス手段123を有している。問合せ要求受信手段124は、端末からのデータベースへの問合せ要求SQL（Structured Query Language）を受信し、受け付けの可否をチェックし、データ処理要求用のトランザクションを生成するものである。

【0032】問合せ最適化手段125は、生成したデータ処理要求用のトランザクションに基づいて、受信したデータベースへの問合せ要求SQLについて構文解析、意味解析を行ない、スキーマファイル133やロックファイル132にある各種関連情報を参照しながら、該問合せ要求に応じたデータベースファイル131等への最適なアクセス経路を決定し、最も実行時間が短いアクセスプラン128を生成するものである。

【0033】ここに、アクセスプラン128は、問合せ最適化手段125により最適化が図られたデータベースアクセス経路に関するデータ処理の実行順序すなわちデータ処理手順を示す内部処理コードを記録しているものであり、問合せ要求SQLの実行手順を示している。アクセスプラン128は、原則として、図2に示すメモリ12上に保存される。

【0034】問合せ実行手段126は、問合せ最適化手段125により生成されたアクセスプラン128に基づいて、ロックファイル132にあるロックモード情報を参照しながら、データベースファイル131内の所望のデータ情報ファイル、あるいは、スキーマファイル133内の所望のスキーマ情報ファイルへのアクセスを行ない、アクセスプラン128に示された問合せ要求に基づいたデータ処理を施して、データ処理結果としての問合せ結果を出力するものである。問合せ結果送信手段127は、問合せ実行手段126が出力した問合せ結果を通信ネットワーク40を介して、要求元の端末へ返送するものである。

【0035】また、データベースファイルアクセス手段121は、問合せ実行手段126からのデータ処理要求種別に応じて、データベースファイル131内の所望のデータ情報ファイルにアクセスし、該データ情報ファイルに関する検索、読み出し、更新、追加あるいは削除等の入出力処理を行なうものである。

【0036】ロックファイルアクセス手段122は、問合せ最適化手段125、あるいは、問合せ実行手段12

6からの要求に基づいて、ロックファイル132内の所望のロック情報ファイルにアクセスし、該ロック情報ファイルに関する検索、読み出し、更新、追加あるいは削除等の入出力処理を行なうものである。

【0037】また、スキーマファイルアクセス手段123は、問合せ最適化手段125、あるいは、問合せ実行手段126からの要求に基づいて、スキーマファイル133内の所望のスキーマ情報ファイルにアクセスし、該スキーマ情報ファイルに関する検索、読み出し、更新、追加あるいは削除等の入出力処理を行なうものである。

【0038】次に、本発明にかかるデータベース管理システムDBMSの概略の動作を図5に示すフローチャートに基づいて説明する。まず、通信ネットワーク40を介して、端末からのデータ処理要求が問合せ要求受信手段124で受信されると、問合せ要求受信手段124は、該データ処理要求の受付処理を行なった後、該データ処理要求用のトランザクションを生成させて、問合せ最適化手段125に該データ処理要求用のトランザクションを送信する（S1）。

【0039】該データ処理要求用のトランザクションを受信した問合せ最適化手段125は、該トランザクション内に含まれている問合せ受信手段124からのデータ処理要求に応じたデータ処理手順を、該データ処理要求に関連するスキーマ情報及びロックモード情報に基づいて生成させ、順次、探索空間に追加していく（S2）。なお、スキーマ情報は、スキーマファイルアクセス手段123を介して、スキーマファイル133内の所望のスキーマ情報ファイルから読み出され、ロックモード情報は、ロックファイルアクセス手段122を介して、ロックファイル132内の所望のロック情報ファイルから読み出される。

【0040】また、問合せ最適化手段125は、生成された探索空間内のデータ処理手順の中から、最短の実行時間及び並行動作の可能性をチェックして、最適のデータ処理手順となるアクセスプラン128を作成し、問合せ実行手段126に送出する（S3）。なお、作成されたアクセスプラン128には、データ処理要求の競合防止用の排他制御を行なうために、データ処理の実行の前後において、データオブジェクト（処理対象のデータ情報ファイルやデータ情報ファイル内のデータレコード）やスキーマオブジェクト（処理対象のスキーマ情報ファイルやスキーマ情報ファイル内のスキーマ情報）に関するロックモードを設定したり、解除したりするデータ処理手順も含まれている。

【0041】問合せ最適化手段125で作成されたアクセスプラン128に基づいて、問合せ実行手段126は、ロックファイルアクセス手段122を介して、処理対象のデータオブジェクトやスキーマオブジェクトの競合状態をチェックし（S4）、競合状態にある場合は（S4のYES）、競合している他のデータ処理が完了

するまで待たせ処理を行なうか、あるいは、要求されたデータ処理を行なうことができない旨の理由を付して、問合せ結果送信手段127に対して拒絶通知を送信する。図5においては、あらかじめ定められた時限のタイミングを待合せた(S10)後、再度競合状態をチェックするためにステップS4に戻る場合の処理を示している。

【0042】一方、前記データオブジェクトや前記スキーマオブジェクトのデータ処理の実行が許容されている状態であれば(S4のNO)、該データオブジェクトや該スキーマオブジェクトに対して、データ処理要求種別に対応した所望のロックモードを設定させるべく、ロックファイルアクセス手段122を介して、ロックファイル132内の指定されたロック情報ファイルに書き込み処理を行なう(S5)。

【0043】その後、問合せ実行手段126は、処理対象の前記データオブジェクトあるいは前記スキーマオブジェクトに対して、アクセスプラン128に記述されているデータ処理を施す(S6)。すなわち、アクセスプラン128に記述されているデータ処理が検索読み出しを指示している場合、前記データオブジェクトあるいは前記スキーマオブジェクトを、データベースファイルアクセス手段121あるいはスキーマファイルアクセス手段123を介して、データベースファイル131あるいはスキーマファイル133の中を検索して読み出す。また、アクセスプラン128に記述されているデータ処理要求が前記データオブジェクトあるいは前記スキーマオブジェクトに対するデータ更新処理を指示している場合は、該データ処理要求に応じたデータ更新処理を前記データオブジェクトあるいは前記スキーマオブジェクトに施した後、データベースファイル131あるいはスキーマファイル133の中にデータベースファイルアクセス手段121あるいはスキーマファイルアクセス手段123を介して書き込む。

【0044】アクセスプラン128に基づくデータ処理が完了した場合、前記データオブジェクトや前記スキーマオブジェクトに設定していたロックモードを解除させるために、ロックファイルアクセス手段122を介して、ロックファイル132内の指定されたロック情報ファイルへの書き込み処理を行なう(S7)。更に、問合せ実行手段126は、アクセスプラン128に基づいたデータ処理手段を実行した結果を、問合せ結果として問合せ結果送信手段127に出力する(S8)。問合せ結果送信手段127は、通信ネットワーク40を介して、要求元の端末に対して、該問合せ結果を送信する(S9)。

【0045】次に、本発明において、データ処理要求の競合防止用に設けたロックモードに関するロック機構について説明する。従来のデータベース管理システムにおいては、前述したように、データ処理要求のトランザク

ションが実行されている(すなわちデータ処理が実行されている)途中においては、データ情報ファイルの構造を定義しているスキーマ情報ファイルが更新されていないこと(すなわち、データ再構成処理がなされていないこと)が前提とされている。本発明においては、データ情報ファイルのみではなく、スキーマ情報ファイルも含むいずれのデータ処理要求のトランザクションが実行されている(すなわち、データ処理が実行されている)途中であっても、スキーマ情報ファイルの変更処理が可能となるように、データ情報ファイルのみでなく、スキーマ情報ファイルに対しても、データ情報ファイルと全く同様のロック機構を導入し、スキーマ情報ファイルの変更処理も一種のデータ処理要求用のトランザクションとして扱うこととしている。かかるロック機構を導入することによって、通常の前記データ情報ファイルへのデータ処理要求と、スキーマ情報ファイルに対するデータ処理要求(スキーマ変更処理要求、すなわち、データ再構成要求)とを同時に受け付けて、同時実行させることができる並行動作性を確保している。

【0046】本発明にかかるデータ処理要求競合防止用のロック機構を実現するためのロックモードとしては、データ情報ファイル用、スキーマ情報ファイル用のそれぞれに対して、X、SIX、S、IX及びISの5つのロックモードを用意する。かかる5つのロックモード間の両立性(Compatibility)を図3に示す。ここに、両立性とは、同一のオブジェクト(すなわち、データ処理要求の対象となるデータ情報ファイルあるいはスキーマ情報ファイル及びデータ情報ファイル内の構成要素であるデータレコードあるいはスキーマ情報ファイル内の構成要素であるスキーマ情報)に対して、複数のデータ処理要求のトランザクションが存在し、同時に実行されんとする場合に、それぞれのトランザクションが指定するロックモードを同時に設定することができるか否か(両立可能か否か)という性質を示すものである。図3において、「Y」は両立可能であることを示し、「N」は両立不可能であることを示している。

【0047】図3において、Xとは、排他モード(Exclusive Mode)であり、該排他モードXが設定されているオブジェクトに対する他からの一切のデータ処理要求(検索、読み取り、更新、追加、削除等)を拒絶するロックモードである。一般的には、データ処理対象のオブジェクトの変更、追加、削除処理を要求するUPDATE文、INSERT文、DELETE文などで指定されているデータ処理を実行する際に設定されるロックモードである。

【0048】また、Sとは、共有モード(Shared Mode)であり、該共有モードSが設定されているオブジェクトに対しては、読み取り動作を行なう他からのデータ処理要求については、受け付けて並行して実行させることができるが、変更、追加、削除等のデータ処理要求に

については、一切拒絶するロックモードである。一般的には、データ処理要求のオブジェクトの検索、読み出し処理を要求するSELECT文などで指定されているデータ処理を実行する際に設定されるロックモードである。ただし、SELECT文の場合であっても、例外的に排他モードXに強制設定させることももちろん可能である。

【0049】また、IXは、インテント排他モード(Intent Exclusive Mode)であり、該インテント排他モードIXが設定されているオブジェクト内の構成要素に対して排他モードXが設定されていることを示すロックモードである。インテント排他モードIXは、データ情報ファイルやスキーマ情報ファイルを構成する表(テーブル)に対して設定されるものであり、表(テーブル)自体に排他モードXが設定されている場合には、該表(テーブル)に対してインテント排他モードIXを設定させる必要はない。

【0050】ISは、インテント共有モード(Intent Shared Mode)であり、該インテント共有モードISが設定されているオブジェクト内の構成要素に対して共有モードSが設定されていることを示すロックモードである。インテント共有モードISは、データ情報ファイルやスキーマ情報ファイルを構成する表(テーブル)に対して設定されるものであり、表(テーブル)自体に共有モードSが設定されている場合には、該表(テーブル)に対してインテント共有モードISを設定させる必要はない。

【0051】SIXは、インテント排他付共有モード(Shared with Intent Exclusive Mode)であり、該インテント排他付共有モードSIXが設定されている表(テーブル)オブジェクトは、一旦、共有モードSに設定されて、データ処理を実行中であったが、該表(テーブル)オブジェクト内のある構成要素に対して排他モードXを設定させるデータ処理が発生したため、該表(テーブル)オブジェクトにインテント排他モードIXをも設定させたいことを示すロックモードである。例えば、大きな表(テーブル)に対して、反復読み取り動作させるSELECT文を実行させて、大きな該表(テーブル)に対して共有モードSを設定している状態にある際に、UPDATE文により、該表(テーブル)の中の構成要素である1つの行(ロー)のみを更新したい要求が発生し、該1つの行(ロー)に対して排他モードXを設定した場合には、該表(テーブル)に対して共有モードSからインテント排他付共有モードSIXに自動的に遷移させて設定し直すことになる。

【0052】すなわち、インテント排他モードIX、インテント共有モードIS及びインテント排他付共有モードSIXは、データ情報ファイルやスキーマ情報ファイルに対する複数のデータ処理要求の同時実行性を向上させるために設けられているロックモードである。例え

ば、特定の表(テーブル)内の構成要素である行(ロー)あるいは列(カラム)のみを更新するデータ処理要求を実行させる場合に、同じ該表(テーブル)内の他の行(ロー)あるいは列(カラム)に対する参照要求の同時実行を許容する際には、該表(テーブル)に対して、インテント排他モードIXが設定され、更新対象となる行(ロー)あるいは列(カラム)に対してのみ排他モードXが設定されることになる。

【0053】また、かかるロックモードは、前述のごとく、データ情報ファイルのみでなく、スキーマ情報ファイルに対しても設定することになる。すなわち、例えば、データ処理対象のオブジェクトがデータ情報ファイルにある場合であっても、データ情報ファイルのみでなく、該データ情報ファイルと関連するスキーマ情報ファイルに対しても、同時にロックモードの設定がなされることにより、データ情報ファイルへのデータ処理とスキーマ情報ファイルへのデータ処理との並行動作を実現させることができる。

【0054】次に、データ情報ファイルあるいはスキーマ情報ファイルに対するデータ処理要求のロック機構の動作について、簡単なデータ情報ファイルとスキーマ情報ファイルとを例にとって、更に詳細に説明する。ここに、図4は、データ処理要求の対象となるデータ情報ファイルすなわちデータオブジェクト及び該データオブジェクトのデータ構造を定義するスキーマ情報ファイルすなわちスキーマオブジェクトの構成の一例を示す図である。

【0055】すなわち、スキーマファイル133内のスキーマ情報ファイルとして、列(カラム)C1、C2についてデータ構造を定義している表Rが存在している。ここで、表R、列(カラム)C1、C2に対応して、それぞれ、スキーマオブジェクトS(R)、S(C1)、S(C2)と表わすこととする。一方、データベースファイル131内のデータ情報ファイルとして、前記スキーマオブジェクトによりデータ構造が定義されているデータ情報ファイルとして表Rが存在し、該データ情報ファイルである表Rは、各行(ロー)毎に列(カラム)C1、C2に対応したデータ属性(アトリビュート)を有するタプル(tuple)情報として、t1、t2、t3の3行(ロー)からなる表(テーブル)を構成している。ここで、該表R、行(ロー)t1、t2、t3に対応して、それぞれ、データオブジェクトD(R)、D(t1)、D(t2)、D(t3)と表わすこととする。

【0056】まず、データ情報ファイルに関するロックモードの設定動作のみに着目して、ロック機構の動作例を説明する。例えば、データ情報ファイルすなわち表Rのデータを、順次、検索していくデータ処理要求トランザクションがある場合、表R全体にロックモードが設定される。かかる場合において、表R全体にかかるデータ処理要求の条件に該当するいずれかの行(ロー)を検索



して、該行（ロー）のデータの更新を伴う場合は、表R全体すなわちデータオブジェクトD（R）に排他モードXが設定される。したがって、該検索処理が終了するまで、表Rへのアクセスを要求する他の一切のデータ処理要求トランザクションは、図3に示すごとく、表RすなわちデータオブジェクトD（R）に対しては、競合状態となるため、図4のステップS4のYESの判定を経て、ステップS10を経由して、再びステップS4に戻る待ち合せ状態となる。

【0057】一方、データファイルすなわち表R全体に関するデータの読み出し動作のみの場合には、表R全体すなわちデータオブジェクトD（R）に対して共有モードSが設定されている状態になる。かかる状態においては、表Rへのアクセスを並行して要求する他のデータ処理要求トランザクションが、表R内の特定の行（ロー）のデータの読み出しのみを行なうデータ処理要求の場合は、該他のデータ処理要求トランザクションの実行時においては、表R全体に対してはISモードが設定され、読み出し対象行（ロー）に対しては、Sモードが設定されることになるので、図3に示すごとく、表RすなわちデータオブジェクトD（R）に対する競合状態は発生していない。したがって、図4のステップS4のNOの判定を経て、両者のデータ処理要求トランザクションとも、同時実行動作が可能となるので、かかる場合においては、データベース処理の高速化を図ることができる。

【0058】また、データ情報ファイルすなわち表Rの特定の行（ロー）t<sub>1</sub>のみにに対してデータ更新を行なうデータ処理要求トランザクションを実行させる場合、表R全体すなわちデータオブジェクトD（R）に対しては、インテント排他モードIXが設定され、データ更新を行なう対象の行（ロー）t<sub>1</sub>すなわちデータオブジェクトD（t<sub>1</sub>）に対しては、排他モードXが設定される。したがって、前記データ更新が終了するまで、図3に示すごとく、行（ロー）t<sub>1</sub>すなわちデータオブジェクトD（t<sub>1</sub>）に対しては、他の一切のデータ処理要求トランザクションが競合状態となり、実行を待ち合せることとなるが、表R内の他の行（ロー）すなわちデータオブジェクトD（t<sub>2</sub>）乃至D（t<sub>3</sub>）に対してアクセスを要求するデータ処理要求トランザクションに対しては、読み出し、書き込みとともに同時実行動作が許容される状態であり、データベース処理の高速化に貢献している。

【0059】更に、データ情報ファイルすなわち表Rの特定の行t<sub>1</sub>のみに対するデータ読み出しを行なうデータ処理要求トランザクションを実行させる場合、表R全体すなわちデータオブジェクトD（R）に対しては、インテント共有モードISが設定され、データ読み出し対象の行（ロー）t<sub>1</sub>すなわちデータオブジェクトD（t<sub>1</sub>）に対しては、共有モードSが設定される。したがって、行（ロー）t<sub>1</sub>の前記データ読み出しが終了するまでは、図3に示すごとく、行（ロー）t<sub>1</sub>すなわち

データオブジェクトD（t<sub>1</sub>）に対してデータ変更を要求するがとき他のデータ処理要求トランザクションは、競合状態となり、待ち合せことになるが、行（ロー）t<sub>1</sub>すなわちデータオブジェクトD（t<sub>1</sub>）に対する読み出し動作については、表Rの他の行（ロー）すなわちデータオブジェクトD（t<sub>2</sub>）乃至D（t<sub>3</sub>）に対するいかなるアクセス動作とともに、他のデータ処理要求トランザクションは同時実行動作が可能となり、更に、データベース処理の高速化が期待できる。

【0060】なお、上述した説明においては、データ情報ファイルに関するロックモードの設定動作のみに着目して説明しているが、かかるデータ情報ファイルに対するアクセス動作を実行させる場合、データ処理の対象となるデータ情報ファイルのデータ構造を定義しているスキーマ情報ファイルに対しても、前述したごとく、同時に、共有モードSおよび／またはインテント共有モードSが設定される。すなわち、対象とするデータ情報ファイルである表R全体に対するデータ処理要求トランザクションを実行させる場合には、該データ情報ファイルである表Rのデータ構造を定義しているスキーマ情報ファイルすなわち表R全体に対応するスキーマオブジェクトS（R）に対して、共有モードSが設定される。一方、前記データ情報ファイルすなわち表R内の特定の行（ロー）に対するデータ処理要求トランザクションを実行させる場合には、スキーマオブジェクトS（R）内にあり、前記データ情報ファイルの行（ロー）t<sub>1</sub>、t<sub>2</sub>、t<sub>3</sub>すなわちデータオブジェクトD（t<sub>1</sub>）、D（t<sub>2</sub>）、D（t<sub>3</sub>）のデータ構造すなわち列（カラム）C1及びC2の構造を定義しているスキーマオブジェクトS（C1）及びS（C2）に対して、共有モードSが設定され、表Rを示すスキーマオブジェクトS（R）に対しては、インテント共有モードISが設定される。

【0061】かかるごとく、データ処理対象のデータ情報ファイルに対応するスキーマ情報ファイルに対しても、同時にロックモードを設定することにより、データ情報ファイルに対するデータ処理要求トランザクションが実行中の状態にあっても、対応するスキーマ情報ファイルの更新処理を競合させることなく並行して動作させることが可能となり、逆に、スキーマ情報ファイルの更新中にデータベースファイルへのアクセス処理を停止させるがとき運用の不便さは回避させることができる。

【0062】次に、スキーマ情報ファイルに関するロックモードの設定動作について、すなわち、スキーマ情報ファイルに対するデータ処理要求トランザクションが発生している場合について説明する。まず、前述の図4に示す表Rのデータ構造を規定しているスキーマ情報ファイルに関し、列（カラム）C1の定義内容を変更せんとするデータ処理要求トランザクションT1が存在している場合、問合せ実行手段126において、定義内容の変更処理が終了するまで、変更処理の対象となるスキーマ

オブジェクトS (C1) に対して排他モードXが設定され、また、該スキーマオブジェクトS (C1) を有する表R対応のスキーマオブジェクトS (R) に対しては、intent排他モードIXが設定される。

【0063】かかるデータ処理要求トランザクションT1が先行して存在している状態において、列(カラム)C1及びC2に定義されている内容を表示させるデータ処理要求トランザクションT2が発発して発生した場合、該トランザクションT2を実行させようとする問合せ実行手段126は、データ処理の対象となるスキーマオブジェクトS (C1) 及びS (C2) に対して、共有モードSを設定させんとし、また、該スキーマオブジェクトS (C1) 及びS (C2) を有する表R対応のスキーマオブジェクトS (R) に対しては、intent共有モードISを設定させんとする。

【0064】しかしながら、図3に示すごとく、スキーマオブジェクトS (C1) に対して設定されている排他モードXは、いかなるロックモードとも両立を認めていないので、両者のトランザクションT1及びT2は、スキーマオブジェクトS (C1) において、競合状態となる。

【0065】したがって、後発のトランザクションT2は、図5に示すフローチャートのステップS4において、競合状態が生じていると判定されて、先発のトランザクションT1による列(カラム)C1の定義変更処理が終了するまで、図5に示すフローチャートのステップS10を経由してステップS4に戻るループを繰り返すことにより、待機状態とされる。

【0066】また、前記のトランザクションT1が先行して存在している状態において、表Rの列(カラム)C2に定義されている内容のみを表示させるデータ処理要求トランザクションT3が発発して発生した場合においては、該トランザクションT3を実行させようとする問合せ実行手段126は、処理の対象となるスキーマオブジェクトS (C2) に対して、共有モードSを設定させんとし、また、該スキーマオブジェクトS (C2) を有する表R対応のスキーマオブジェクトS (R) に対しては、intent共有モードISを設定させんとする。

【0067】かかる状態においては、図3に示すごとく、双方のトランザクションT1とT3との間には競合状態は発生していないので、後発のトランザクションT3も、図5に示すフローチャートのステップS4において、競合状態ではないと判定され、先発のトランザクションT1と同時に実行の動作が可能となる。したがって、両者の同時実行により、データベース処理の高速化を図ることができる。

【0068】なお、前記のトランザクションT1のように、データ情報ファイルすなわち表Rのデータ構造を定義するスキーマオブジェクトを変更させるデータ処理要求トランザクションを実行する場合、該スキーマオブ

ジェクトで定義されているデータ情報ファイルすなわち表Rを示すデータオブジェクトD (R) 側に対して、いかなるロックモードも設定する必要はない。しかしながら、前記のトランザクションT2あるいはT3のように、表Rのデータ構造を定義するスキーマオブジェクトを変更せずに、表示させるようなトランザクションを実行する場合においては、スキーマオブジェクトへのロックモードの設定が完了した後、該スキーマオブジェクトによって、データ構造が定義されているデータ情報ファイルすなわち表Rを示すデータオブジェクトD (R)、及び、該表Rの構成要素である各行(ロー)を示すデータオブジェクトD (t1)、D (t2)、D (t3) に対して、データ処理の競合を防止するために、ロックモードを設定しなければならない。すなわち、前記のトランザクションT2あるいはT3が実行されて、各スキーマオブジェクトS (R)、S (C1)、S (C2) あるいはS (R)、S (C2) へのロックモードの設定が完了した後、該スキーマオブジェクトで定義されている表Rの各行(ロー)を示すデータオブジェクトD (t1)、D (t2)、D (t3) に対して、共有モードSが設定され、表Rを示すデータオブジェクトD (R) に対しては、intent共有モードISが設定される。

【0069】したがって、例えば、前記のトランザクションT2が先発して処理されている状態において、スキーマオブジェクトS (C2) でデータ構造を示すデータオブジェクトD (t1)、D (t2)、D (t3) のデータ内容を更新するデータ処理要求トランザクションT4が発発して発生した場合においては、該トランザクションT4を実行させようとする問合せ実行手段126は、スキーマオブジェクトS (C2) で定義されているデータ情報ファイルすなわち表Rの各行(ロー)を示すデータオブジェクトD (t1)、D (t2)、D (t3) に対して、排他モードXを設定させんとし、表Rを示すデータオブジェクトD (R) に対しては、intent排他モードIXを設定させんとする。更には、前記トランザクションT4の処理対象であるデータ情報ファイルの各行(ロー)内の列C2に対応するスキーマオブジェクトS (C2) に対しても、共有モードSを設定させ、該スキーマオブジェクトS (C2) を有する表R対応のスキーマオブジェクトS (R) に対して、intent共有モードISを設定させんとする。

【0070】かかる状態において、スキーマオブジェクトに対する両トランザクションT2とT4との競合関係についてみれば、トランザクションT2に関しては、スキーマオブジェクトS (C1) 及びS (C2) には共有モードSが、また、スキーマオブジェクトS (R) に対しては、intent共有モードISが設定されており、一方、トランザクションT4に関しては、スキーマオブジェクトS (C2) には共有モードSが、また、スキーマオブジェクトS (R) に対しては、intent共有モ

emnt Language) によるデータ処理、すなわち、データオブジェクトに対する検索、読み出し、変更、追加、削除等のデータ処理との双方において、スキーマオブジェクトとデータオブジェクトとの双方に対するロックモードを設定しながら、データ処理を行なわしめることにより、かかるデータ処理の並行動作性を確保するとともに、直列化可能性も保証している。

【0078】ここで、直列化可能性とは、複数個のデータ処理要求トランザクションに関し、並行させてデータ処理を実行させて得られたデータベースの状態が、複数個の該データ処理要求トランザクションを、ある順序に従って逐次実行させて得られるデータベースの状態と一致するトランザクションの状態と一致するデータ処理の性質を示している。かかる直列化可能性が保証されると、ある一貫性を有するデータベースに対して、複数個のデータ処理要求トランザクションを並行させて実行させても、新たな一貫性を有するデータベースの状態に移移させることを保証することができる。

【0079】また、本発明にかかる問合せ最適化手段125においては、スキーマオブジェクトに対して設定されているロックモードの状態に基づいて、問合せ実行手段126で実行される最適の処理手順を示すアクセスプラン128の内容を変更させることもできる。例えば、問合せ最適化手段125が、新たなデータ処理要求に応じたアクセスプラン128を生成せんとしている時点において、データベースの索引関係の再構成を行なうスキーマ情報変更のデータ処理要求トランザクションが実行されている状況になっている場合、かかる索引情報変更に関するスキーマ情報に対するロックモードも有しているので、データ処理の並行動作性を考慮して、前記索引関係の再構成に係るデータを用いないアクセスプラン128を生成させることも可能である。

【0080】更には、問合せ機能を更に拡張させた機能であるトリガやアサーション機能に基づくデータ処理要求（データオブジェクトあるいはスキーマオブジェクトに対する検索、読み出し、変更、追加、削除等のデータ処理要求）に対応して、データオブジェクトのみでなく、関連するスキーマオブジェクトに対するロックモードを設定させることにより、トリガやアサーション機能に関連するスキーマオブジェクトや該スキーマオブジェクトにより定義付けられているデータオブジェクトに対するデータ処理要求（検索、読み出し、変更、追加、削除等のデータ処理要求）トランザクションとの並行動作性も実現させることができる。ここに、トリガやアサーション機能とは、あらかじめ定められた特定の条件が成立した場合に、あらかじめ定められているデータ処理手順を自動的に起動させるようにした機能であり、例えば、給与データの一部を変更した場合、自動的に全員の給与の平均データを算出して設定し直させる動作を起動

させるような機能である。

#### 【0081】

【発明の効果】データベースを構成する各データ情報ファイルのデータ構造を定義しているスキーマ情報に対するロックモードを設けることにより、データ処理の要求に関するトランザクションの実行途中（すなわちデータ処理の実行途中）であっても、該データ処理にかかるデータ情報ファイルに関するスキーマ情報の変更を並行して行なわせることが可能であり、同時実行可能なデータ処理要求数を増加させることができるので、データベース管理システム全体の高速性、高処理効率を達成させることができる。

【0082】また、スキーマ情報の変更時において、データベース管理システムの運用停止を回避させることができるので、データベース管理システムの運用手順を簡素化を図ることができ、運用コストを低減することができる。

#### 【図面の簡単な説明】

【図1】 本発明にかかるデータベース管理システムDBMSを構成する各種の処理と、該処理の処理対象となるデータベースDBを構成する各種の情報ファイルとを示すブロック構成図である。

【図2】 本発明の一実施例にかかるデータベースシステムのハードウェア構成を示すブロック構成図である。

【図3】 本発明にかかるロック機構を実現するロックモードの両立性を示すマトリクス図である。

【図4】 データ処理要求の対象となるデータオブジェクト及びスキーマオブジェクトの構成の一例を示す図である。

【図5】 本発明にかかるデータベース管理システムDBMSの概略の動作を説明するためのフローチャートである。

#### 【符号の説明】

10…サーバ、11, 21, 31…CPU、12, 22, 32…メモリ、13, 23, 33…ハードディスク、14, 24, 34…キーボード、15, 25, 35…ポインティングデバイス、16, 26, 36…ディスプレイ、20, 30…端末1, 端末N、40…通信ネットワーク、120…データベース管理システムDBMS、121…データベースファイルアクセス手段、122…ロックファイルアクセス手段、123…スキーマファイルアクセス手段、124…問合せ要求受信手段、125…問合せ最適化手段、126…問合せ実行手段、127…問合せ結果送信手段、128…アクセスプラン、130…データベースDB、131…データベースファイル、132…ロックファイル、133…スキーマファイル、221, 321…問合せ要求入力手段、222, 322…問合せ結果表示手段。

ードISが設定されんとしている。図3に示すごとく、両方のトランザクションT2とT4との間には、競合関係は、発生せず、同時実行動作が可能である。

【0071】しかしながら、データオブジェクトに対する両方のトランザクションT2とT4との競合関係をみると、トランザクションT2の実行時においては、データオブジェクトD(t<sub>1</sub>)、D(t<sub>2</sub>)、D(t<sub>3</sub>)に共有モードSが、また、データオブジェクトD(R)に\_intent共有モードISが設定されている。一方、トランザクションT4に関しては、データオブジェクトD(t<sub>1</sub>)、D(t<sub>2</sub>)、D(t<sub>3</sub>)に対して、排他モードXが、また、データオブジェクトD(R)に対しては、intent排他モードIXが設定されんとするため、図3に示すごとく、両方のトランザクションT2とT4とは、競合関係となる。したがって、後発のトランザクションT4は待機状態とされることになる。

【0072】次に、図4に示す表Rの構造を規定しているスキーマ情報として、列(カラム)C1に索引Iが定義されていて、索引Iにより、表Rが他の表と関係付けられている場合について説明する。該索引Iの再構成(作り直し)を行うスキーマ情報変更のデータ処理要求トランザクションT5が発生した場合においては、索引Iを示すスキーマオブジェクトS(I)に対しては、排他モードXが設定され、表Rに対応するスキーマオブジェクトS(R)には、intent排他モードIXが設定されることになる。

【0073】ここで、表Rで示すデータ情報ファイルの列(カラム)C1に格納されているデータをSELECT文などで検索するようなデータ処理要求トランザクションが同時に発生した場合においては、列(カラム)C1に格納されているデータの変更は伴わないので、スキーマオブジェクトS(I)に対して、ロックモードを設定する必要はない。

【0074】しかしながら、表Rで示すデータ情報ファイルの列(カラム)C1に格納されているデータをUPDATE文などで変更するようなデータ処理要求トランザクションが発生した場合においては、スキーマオブジェクトS( )に対しても共有モードSを設定させなければならない。すなわち、スキーマ情報として索引Iが再構成されんとしている場合に、データ情報ファイルの列(カラム)C1に格納されているデータを変更せんとするデータ処理要求トランザクションが発生した場合には、スキーマオブジェクトS(I)に対する競合の発生を検出させて、索引データである列(カラム)C1のデータ変更の待合せ処理を行なわせることにより、両者のトランザクションの並行同時を可能とするものであり、索引Iの再構成後に、再度データ情報ファイルの列(カラム)C1のデータ変更を実行させなければならないような事態の発生を防止することができる。

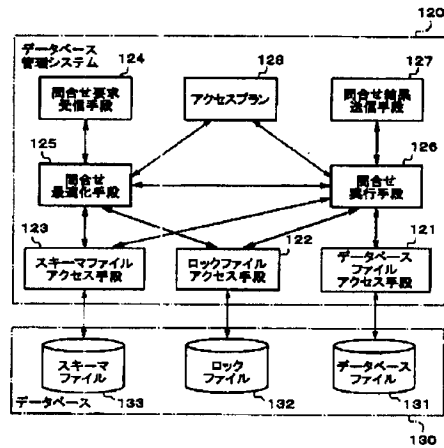
【0075】以上に述べたごとく、本発明にかかるデー

タベース管理方法及びデータベース管理装置においては、データオブジェクトのみでなく、スキーマオブジェクトに対しても、競合防止用のロックモードを設定させることにより、データベースに関するデータ処理の並行動作性を維持し、直列化可能性を保証している。すなわち、例えば、スキーマ情報を変更するデータ処理要求トランザクションが発生した場合、変更対象となるスキーマオブジェクトに対してロックモードを設定する。この場合、変更対象がデータの属性を規定する列(カラム)である場合は、該列(カラム)に対して、排他モードXを、また、該列(カラム)が構成要素となっている表(テーブル)に対しては、intent排他モードIXを設定する。また、変更対象が表全体に関わる場合は、変更対象となる表全体に対して、排他モードXを設定するが、該表の構成要素である各列(カラム)に対しては、排他モードXを設定する必要はない。なお、索引が定義されている場合、索引を示すスキーマオブジェクトに対しても、変更を必要とする場合には、排他モードXを設定する。また、スキーマ情報を変更するデータ処理要求トランザクションである場合、該スキーマ情報によってデータ構造が定義されているデータオブジェクトに対しては、ロックモードを設定する必要はない。

【0076】一方、スキーマ情報を変更せずに、参照のみのデータ処理要求トランザクションが発生した場合、参照対象となるスキーマオブジェクトに対して、ロックモードを設定する。この場合、参照対象がデータの属性を規定する列(カラム)あるいは索引である場合は、該列(カラム)あるいは索引に対して共有モードSを、また、該列(カラム)あるいは索引が構成要素となっている表(テーブル)に対しては、intent共有モードISを設定する。更に、該スキーマ情報によって、データ構造が定義されているデータオブジェクトに対してもロックモードを設定する必要がある。すなわち、スキーマオブジェクトに対するロックモードの設定が完了した時点で、対応するデータオブジェクトに対してもロックモードを設定する。ここで、該データオブジェクトである表(テーブル)に対しては、intent共有モードISまたは共有モードSを、また、該データオブジェクトの表の構成要素となる各行(ロー)に対しては、共有モードSを設定する。

【0077】すなわち、本発明にかかるデータベース管理方法及びデータベース管理装置においては、データ情報ファイルの構造を定義するスキーマ情報をスキーマ情報ファイルとして生成させるデータ定義言語(DDL: Data Definition Language)によるデータ処理、すなわち、スキーマオブジェクトに対する検索、読み出し、変更、追加、削除等のデータ処理と、該スキーマ情報ファイル内のスキーマ情報によりデータ構造が定義されているデータ実体を備えているデータ情報ファイルに対するデータ操作を行うデータ操作言語(DML: Data Manag

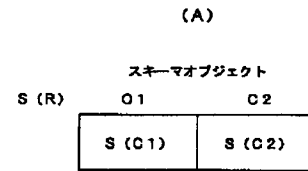
【図 1】



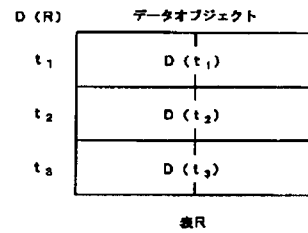
【図 3】

	X	SIX	S	IX	IS
X	N	N	N	N	N
SIX	N	N	N	N	Y
S	N	N	Y	N	Y
IX	N	N	N	Y	Y
IS	N	Y	Y	Y	Y

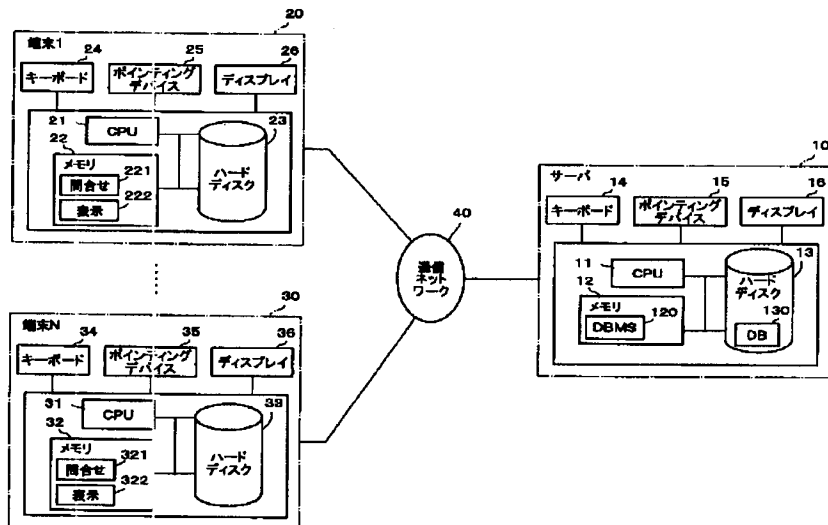
【図 4】



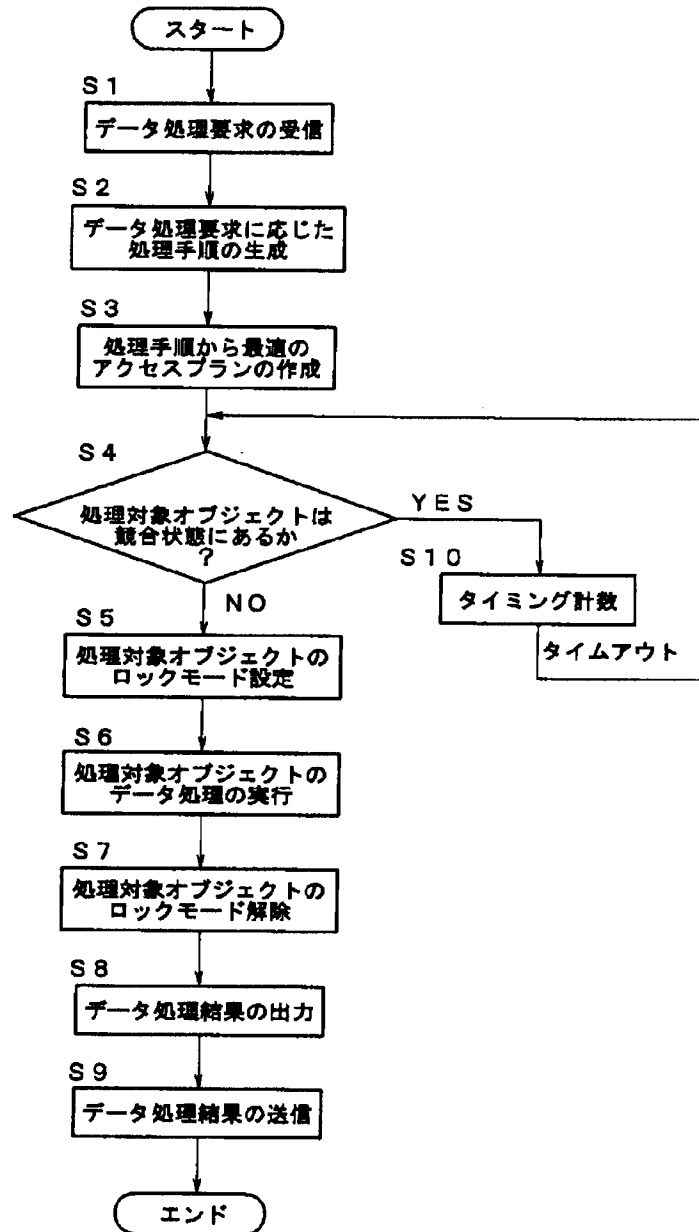
(8)



【図 2】



【図5】



フロントページの続き

(72)発明者 平岡 卓也  
東京都大田区中馬込1丁目3番6号 株式  
会社リコー内

(72)発明者 池田 哲也  
東京都大田区中馬込1丁目3番6号 株式  
会社リコー内

(72)発明者 前田 薫  
東京都大田区中馬込1丁目3番6号 株式  
会社リコー内

Fターム(参考) 5B075 NR02 PP02 PP03 PQ02 QS07  
QT06  
5B082 AA01 GA03 GA07 GA08 GA15